Understanding Innovation

Hasso Plattner
Christoph Meinel
Larry Leifer *Editors*

Design Thinking Research

Taking Breakthrough Innovation Home

Springer

Understanding Innovation

Series editors
Christoph Meinel
Larry Leifer

More information about this series at http://www.springer.com/series/8802

Hasso Plattner • Christoph Meinel • Larry Leifer
Editors

Design Thinking Research

Taking Breakthrough Innovation Home

Springer

Editors
Hasso Plattner
Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany

Christoph Meinel
Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany

Larry Leifer
Stanford University
Stanford, CA, USA

ISSN 2197-5752 ISSN 2197-5760 (electronic)
Understanding Innovation
ISBN 978-3-319-40381-6 ISBN 978-3-319-40382-3 (eBook)
DOI 10.1007/978-3-319-40382-3

Library of Congress Control Number: 2016943632

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

Design thinking is spreading. More and more people and organizations experience the positive impact of design thinking on their innovation culture and output. And more and more educational institutions integrate the approach into their curriculum or even implement their own schools of design thinking following the role model of the d.schools in Stanford and Potsdam. An entrepreneurial and user-centered mindset is nurtured here, which is fundamental for innovation. Taking this approach into companies stimulates corporate entrepreneurial thinking and acting—fueling a real *intra*preneurial culture. This is vital especially for big corporations as it consequently spurs innovation and change inside these organizations. In promoting creative confidence, curiosity, collaboration, and a trial and error attitude, design thinking fosters this essential cultural change.

The rise of design thinking has increased the desire to better understand how and why it works. Design thinking changes the way people and companies innovate. With design thinking, the innovation process is dynamic, iterative, reflective, fast, tangible, and above all human centered. Those who adopt and assimilate the method can become sustainably innovative and able to design fundamental new products, services, or processes. But how does this effect come about? What is the reason behind the power of design thinking? How does this method work—and why does it sometimes fail? What are the preconditions and impacts?

These are questions that inevitably arise after you come into contact with this method. Curiosity led me to initiate and fund the HPI-Stanford Design Thinking Research Program in 2008. Since then, scientists from the Hasso Plattner Institute in Potsdam, Germany, and the Stanford University, USA, have conducted dozens of research projects on various design thinking-related topics, shedding light on the underlying principles and coming up with new tools and approaches. Over the past years, many interesting and valuable insights have been gained. The findings not only contribute to the academic advancements and ongoing discourse but are meant to provide guidance and inspiration for practitioners and to contribute to the general understanding and, ultimately, to the dissemination of this method—which in the end fosters economic and social innovation.

Despite these advances, there is so much more to find out about the innovation process. We are at a very interesting point in time in the evolution of design thinking, with a growing number of people and organizations that adapt this method and with the equally increasing desire to better understand it. The mandate for research is clear and the field for exploration broad. Seeing the scientific progress made so far, I am pleased to continue my support for the HPI-Stanford Design Thinking Research Program. I am looking forward to the insights that will be gained in the next years and invite you to dive into the latest findings presented in this publication.

Palo Alto, CA Hasso Plattner
Winter 2015/2016

Contents

Design Thinking for the Twenty-First Century Organization 1
Larry Leifer and Christoph Meinel

Thisisdesignthinking.net: A Storytelling-Project 13
Eva Köppen, Jan Schmiedgen, Holger Rhinow, and Christoph Meinel

Part I Design Thinking in Practice

Colliding Influences 19
Holger Rhinow and Christoph Meinel

Mapping and Measuring Applications of Design Thinking in Organizations 35
Adam Royalty and Bernard Roth

The Design Thinking Methodology at Work: Capturing and Understanding the Interplay of Methods and Techniques 49
Thomas Beyhl and Holger Giese

On Creating Workspaces for a Team of Teams: Learnings from a Case Study 67
Marie Klooker, Claudia Nicolai, Stephan Matzdorf, Arne Trost, Karen von Schmieden, Lilith Böttcher, and Ulrich Weinberg

Part II Exploring Human-Technology Interaction

Design Thinking in Health IT Systems Engineering: The Role of Wearable Mobile Computing for Distributed Care 87
Lauren Aquino Shluzas, Gabriel Aldaz, David Pickham, and Larry Leifer

Redesigning Medical Encounters with Tele-Board MED 101
Anja Perlich, Julia von Thienen, Matthias Wenzel, and Christoph Meinel

Embodied Design Improvisation for Autonomous Vehicles 125
David Sirkin, Sonia Baltodano, Brian Mok, Dirk Rothenbücher, Nikhil Gowda, Jamy Li, Nikolas Martelaro, David Miller, Srinath Sibi, and Wendy Ju

Part III Prototyping

Can Anyone Make a Smart Device?: Evaluating the Usability of a Prototyping Toolkit for Creative Computing 147
Joel Sadler, Lauren Aquino Shluzas, Paulo Blikstein, and Sakti Srivastava

Making Examples Tangible: Tool Building for Program Comprehension 161
Marcel Taeumel and Robert Hirschfeld

Case Studies on End-User Engagement and Prototyping during Software Development 183
Franziska Dobrigkeit, Sebastian Meyer, and Matthias Uflacker

Part IV Developing DT Teaching and Coaching Tools and Approaches

Design Thinking At Scale: A Report on Best Practices of Online Courses 217
Mana Taheri, Thomas Unterholzer, and Christoph Meinel

Designing Scalable and Sustainable Peer Interactions Online 237
Chinmay Kulkarni, Yasmine Kotturi, Michael S. Bernstein, and Scott Klemmer

Developing Instrumentation for Design Thinking Team Performance 275
Neeraj Sonalkar, Ade Mabogunje, Halsey Hoster, and Bernard Roth

The Topic Markup Scheme and the Knowledge Handling Notation: Complementary Instruments to Measure Knowledge Creation in Design Conversations 291
Axel Menning, Andrea Scheer, Claudia Nicolai, and Ulrich Weinberg

Developing Novel Neuroimaging Paradigm to Assess Neural Correlates of Improvisation and Creative Thinking Using fMRI 309
Manish Saggar, Lindsay C. Chromik, Adam Royalty, Grace Hawthorne, and Allan L. Reiss

Erratum: Design Thinking Research E1

Design Thinking for the Twenty-First Century Organization

Larry Leifer and Christoph Meinel

1 How Does Organizational Mismatch Impact Design Thinking and Innovation?

Given

"Team of teams" organizations are good for innovation.
"Command-Control" organizations are good for efficiency.

Grand Challenge

How does one interface these highly disparate organizations to preserve and promote collective creativity?

The probability of breakthrough innovation has increased as we gain a deeper understanding of design innovation processes and the organizations that use them best. It is increasingly clear that an organizational "impedance mismatch" is a barrier to bringing breakthrough innovation home to corporations, governments, and economies.

A brief working definition of **"impedance mismatch"** is that which inhibits the movement of electrons, protons, money, and ideas. We include especially the inhibition of free flowing human communication and creative experimentation.

L. Leifer
Stanford Center for Design Research, Stanford University, Panama Mall 424, Stanford 94305-2232, CA, USA

C. Meinel (✉)
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: meinel@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_1

Design research has begun to tackle this issue with new metrics and heightened awareness of organizational structure as a core barrier to growth through innovation.

Discovered

A "team of teams" organization is good for design thinking and breakthrough innovation.

The industry sponsored engineering design project course at Stanford University, ME310-Global, has evolved over the past 30 years to function as a global team of teams. Companies bring engineering design-innovation challenges to the course each year with project briefs that typically take the form "re-invent X."

As the curriculum evolved, the traditional notion of isolated teams working on "design challenges" gave way to evidence that teams helping each other across design challenges, corporate identities, and personal relationships were outperforming locally insular teams. And then the curriculum began to "spin-off" to other universities and their networks of companies and colleagues. In time the outer constellation of industry sponsored design-X challenges became known as the SUGAR Network. An imaginative visualization of the network, Fig. 1, has become an iconic representation of a human-centric breakthrough innovation challenged team of teams. Two different academic teams of 3–4 graduate students at

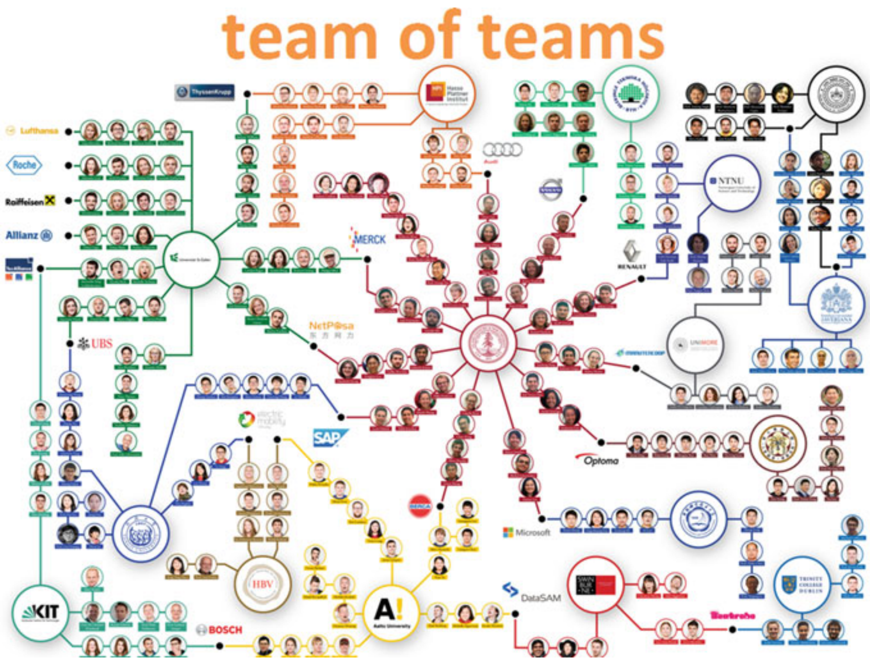


Fig. 1 The 310-SUGAR Network (2014–2015) is an academic-industry team of teams in open association of students, faculty, and corporate representatives driven to deliver breakthrough innovation for business defined design challenges

internationally dispersed universities own each project. The university constellation is aided by corporate teams of 3–4 people to further the reach and depth of the team of teams.

The Hasso Plattner Institute (HPI) is a leading member of the network with outreach to its many global initiatives.

Defining characteristics of the twenty-first century innovation ecosystem include the following (after Dorst 2015). How might we help corporations, universities, and societies to accelerate innovation in ways that keep pace with these challenges?

OPEN: unbound systems
COMPLEXITY: systems of system elements and relationships
DYNAMIC: change with diminished element and process half-life
NETWORKED: across organizational elements and relationships

Looking across the abyss between the innovation oriented team of teams towards the efficiency oriented command-control organization we speculate that one needs to see an "intrapreneur" in corporate structure. Speculating further, we hypothesize that the insider intrapreneur needs to be part of an insider "team-of-intrapreneur-teams." The impedance mismatch needs to be matched to an augmented flow structure. This scenario has been forecast by Beth Altringer, November 2013, in her Harvard Business Review article "A New Model for Innovation in Big Companies."

In Altringer's paper she reports that "Studies show that efforts to stimulate intrapreneurship—entrepreneurship within an established company—more often than not fall flat. According to my current research at Harvard on innovation models in global companies across diverse sectors, these types of projects fail between 70 % and 90 % of the time."

Is it time to formalize the role of institutional intrapreneurs?

A formal definition of entrepreneur and entrepreneurship follows:

> **Entrepreneurship** is the process of designing, launching, and running a new business, i.e. a startup company offering a product, process or service. It has been defined as the "...capacity and willingness to develop, organize, and manage a business venture along with any of its risks in order to make a profit." The **entrepreneur** is "a person who organizes and manages any enterprise, especially a business, usually with considerable initiative and risk." "[R]ather than working as an employee, [an entrepreneur] runs a small business and assumes all the risk and reward of a given business venture, idea, or good or service offered for sale. The entrepreneur is commonly seen as a business leader and innovator of new ideas and business processes."
>
> Entrepreneurs perceive new business opportunities and they often exhibit positive biases in their perception (i.e., a bias towards finding new possibilities and unmet market needs) and a pro-risk-taking attitude that makes them more likely to exploit the opportunity. "Entrepreneurial spirit is characterized by innovation and risk-taking".[1]

[1] https://en.wikipedia.org/wiki/Entrepreneurship

Intrapreneurship is the act of behaving like an entrepreneur while working within a large organization. Intrapreneurship is known as the practice of a corporate management style that integrates risk-taking and innovation approaches, as well as the reward and motivational techniques, that are more traditionally thought of as being the province of entrepreneurship.

Pinchot (1984) defined intrapreneurs as "dreamers who do. Those who take hands-on responsibility for creating innovation of any kind, within a business". In 1992, *The American Heritage Dictionary* acknowledged the popular use of a new word, intrapreneur, to mean "A person within a large corporation who takes direct responsibility for turning an idea into a profitable finished product through assertive risk-taking and innovation". Koch (2014) goes further, claiming that intrapreneurs are the "secret weapon" of the business world. Based on these definitions, being an intrapreneur is considered to be beneficial for both intrapreneurs and large organisations. Companies support intrapreneurs with finance and access to corporate resources, while intrapreneurs create innovation for companies.

The intrapreneur is not to be confused with the "innerpreneur", a person who aims at personal fulfilment more than at economic gains when creating a business.[2]

The intrapreneur is driven by most of the same beliefs as the entrepreneur ("there has to be a better way"). Unlike the typical "outsider" entrepreneur, the intrapreneur is part of the organization. There is even a case for the "Chief Intrapreneur", but this title goes against the team of teams organization's values and methods. Perhaps the intrapreneur is chief of the bottom-up brigade; more of an inspiration than a chief.

Institutionalizing the intrapreneur is a move forward with the professionalization of design thinking. Imagine a pan-disciplinary doctoral program for understanding intrapreneurship as design thinking within institutional practice. There is movement in this direction at Stanford University and the Hasso Plattner Institute in Potsdam. The international SUGAR Network and the HPI network of d.schools are both well positioned to implement intrapreneurship training, practice, and needed research.

Once again, the design-paradigm is worth your attention.

We continue to improve our understanding of design thinking, to discover impactful practices, and to disseminate these practices through publications, simulations, emulations, and workshops. The understanding we derive from the study of human teamwork with IT augmentation is, again, foundational.

2 The HPI-Stanford Design Thinking Research Program

With the progressive dissemination of design thinking in practice, education, and academia over the last years, the demand to understand this method has increased. Already back in 2008 the joint HPI-Stanford Design Thinking Research Program was established, funded by the Hasso Plattner Foundation. Within this program,

[2]https://en.wikipedia.org/wiki/Intrapreneurship

scientists from the Hasso Plattner Institute for Software Systems Engineering in Potsdam, Germany, and from Stanford University, USA, strive to gain a deep understanding of the underlying principles of design thinking and, consequently, how and why this innovation method succeeds or fails.

2.1 Program Vision and Goals

Multidisciplinary research teams from HPI and Stanford with backgrounds in disciplines such as engineering, design, humanities or social sciences scientifically investigate innovation and design thinking in all its holistic dimensions. These areas include technical, economic, and human factors. Applying rigorous academic methods, the researchers examine how the innovative process can be improved and further developed.

The program pursues the goal to advance design thinking theory and knowledge within the research community and ultimately improve design practice and education by gathering scientific evidence to support design activities. Beyond a mere descriptive understanding, this program aims, for example, to develop metrics that allow assessment and prediction of team performance to facilitate real-time management of how teams work. Scientists study the complex interaction between members of multi-disciplinary teams, with special regard to the necessity of creative collaboration across spatial, temporal, and cultural boundaries. They design, develop, and evaluate innovative tools and methods that support teams in their creative work. The projects pursue the common questions of why structures of successful design thinking teams differ substantially from traditional corporate structures and how design thinking methods mesh with traditional engineering and management approaches.

Researchers are especially encouraged to develop ambitious, long-term explorative projects that integrate technical, economical, as well as psychological points of view using design thinking tools and methods. Field studies in real business environments are considered especially important to assess the impact of design thinking in organizations and if any transformations of the approach may be warranted.

Special interest lies in the following guiding questions:

- What are people really thinking and doing when they are engaged in creative design innovation?
- How can new frameworks, tools, systems, and methods augment, capture, and reuse successful practices?
- What is the impact of design thinking on human, business, and technology performance?
- How do the tools, systems, and methods really work to create the right innovation at the right time? How do they fail?

Over the past years dozens of research projects have been conducted, our understanding of this field has advanced and new insights and tools have become available. These findings are not only meant to be discussed within the scientific community. With this book they are made known to the public at large and to all who want and need to drive innovation, be it in companies or society.

2.2 Road Map Through This Book

In the seventh program year scientists from HPI and Stanford University have again conducted various research projects on design thinking. Their results are compiled in this book, divided into four sections that illustrate the numerous facets of design thinking.

Design thinking is adopted by more and more people and organizations—in diverse and individual ways. Part 1 "***Design Thinking in Practice***" takes a closer look at how this method is applied in organizations and how it impacts them (e.g., with regard to team interactions or management). Furthermore, a tool is presented that accurately describes how design thinking is applied. The different characteristics of design thinking and what they mean are important to know for practitioners and have therefore been investigated and described, too. The last chapter explores how spaces for innovation teams are created in organizations.

With the technological progress, new opportunities as well as challenges in design processes arise. Therefore, "***Exploring Human-Technology Interaction***" stands in the focus of the book's second part. How new mobile computing devices are able to influence behavior change is examined in the first chapter, which illustrates an application of design thinking in healthcare. With *Tele-Board Med* researchers not only developed a medical documentation system and collaborative eHealth application but also investigated the impact of such a tool on team interactions and feelings. This was done specifically in a therapy context. Furthermore, in three studies researchers describe an embodied design improvisation methodology that is effective in designing the behaviors and interfaces of autonomous vehicles. They thereby look closer at the conceptual phase of design thinking as well as prototyping.

The third part of the book dives deeper into the "***Prototyping***" phase of design thinking. It explores how technical novices can be supported in electronics prototyping. In addition, research also investigates prototyping possibilities in programming, introducing a tool that increases tangibility. Finally, one project provides us insights into the development process in software companies and presents an overview of current practices concerning end-user involvement and prototyping.

The last part of the book is about "***Developing Design Thinking Teaching and Coaching Tools and Approaches.***" Special emphasis is placed on online approaches: researchers investigate whether and how MOOCs are suited for design thinking education. Furthermore, scientists demonstrate how large classes can leverage their scale to encourage mastery through rapid feedback and revision. Projects also address "analogue" team work, providing specific diagnostic instruments based on a visual notation for augmenting design team performance. Finally, researchers investigate the underlying neurocognitive foundation and sustainability of creative capacity enhancement.

2.3 Part I: Design Thinking in Practice

In "**Colliding Influences—When Self-Organizing Teams Encounter Strategic Objectives and Established Routines**" Holger Rhinow and Christoph Meinel illustrate findings from a case study on the impact of design thinking within a large organization. As several teams begin to apply design thinking as a framework for product discovery and development, a growing influence of self-organizing teamwork and the user as a source of inspiration becomes apparent. This stands in contrast to other frameworks for product development within the organization (e.g. Waterfall and Scrum). These new influential factors are to some extent seemingly in collision with other existing influential factors, such as established routines in project management and a corporate strategy. This case study empirically clarifies the impression from previous research that the integration of design thinking appears to be a managerial challenge yet to be mastered.

With the ongoing dissemination of design thinking it is critical to develop tools that accurately describe how the method is being applied in teams and across an organization as a whole. In "**Mapping and Measuring Applications of Design Thinking in Organizations**" Adam Royalty and Bernard Roth introduce two tools in development to meet these goals. The first is an "ecology mapping" that portrays an organization's internal design thinking strategy. The second is a weekly "snapshot" of design thinking activities performed by industry teams working on creative projects.

The design thinking methodology suggests a repertoire of design phases, design activities, and design methods that can be used to solve wicked problems in terms of innovative solutions. However, since it does not prescribe any order of design phases, activities and techniques, their applications lead to different shapes of the design thinking methodology in practice. The authors of "**The Design Thinking Methodology at Work: Capturing and Understanding the Interplay of Methods and Techniques**", Thomas Beyhl and Holger Giese, hypothesize that these shapes of design thinking at work consist of different characteristics depending on the kind of design project that has been conducted. Understanding these characteristics, their influence on the design flow itself, as well as their impact on the outcome of the design project is of major interest to managers, innovators, and researchers. The article reports on the result of a case study that has been conducted to investigate different shapes of the design thinking methodology in practice.

With "**On Creating Workspaces for a Team of Teams—Learnings from a Case Study**" Marie Klooker, Stephan Matzdorf, Claudia Nicolai, Lilith Böttcher, Arne Trost, and Karen von Schmieden offer first insights into defining strategic intent for the development of so-called creative workspaces. On an academic level, previous research has mostly focused on established physical environmental structures, disregarding the contextual level of strategic intent. On a practical level, companies too often copy best practice examples of other innovation labs. Based on the qualitative case study of an organization currently implementing an innovation lab, this chapter introduces a collection of categories defining strategic intent preceding the establishment of innovation laboratories within an organization.

2.4 Part II: Exploring Human-Technology Interaction

In "**Design Thinking in Health IT Systems Engineering: The Role of Wearable Mobile Computing for Distributed Care**" Lauren Aquino Shluzas, Gabriel Aldaz, David Pickham, and Larry Leifer examine the capabilities and boundaries of a hands-free mobile augmented reality (AR) system for distributed healthcare. They use a developer version of the Google Glass™ head-mounted display (HMD) to develop software applications to enable remote connectivity in the healthcare field, and to characterize system usage, data integration, and data visualization capabilities. In this chapter they summarize findings from the assessment of the SnapCap System for chronic wound photography, and present a pilot study. This work contributes to the future implementation of new features aimed at enhancing the documentation and assessment of chronic wounds. It provides insight into the need for future IT systems engineering projects with the goal of improving healthcare connectivity for distributed care.

The path to a satisfying health care outcome is manifold, and the quality of the relationship between patient and health care provider is an impactful factor. In "**Redesigning Medical Encounters with Tele-Board MED**" Anja Perlich, Julia von Thienen, Matthias Wenzel, and Christoph Meinel discuss different models for the classification of patient-provider interaction as well as for patient empowerment. On this theoretical basis, they elaborate how patient-provider interaction can be enhanced in practice by means of the medical documentation system—Tele-Board MED. It is a collaborative eHealth application designed to support the interaction between patient and provider in clinical encounters. Simultaneously, it aims at making case documentation more efficient for providers and more valuable for patients. As a research paradigm, the Tele-Board MED project has used a design thinking approach to understand and support fundamental stakeholder needs. Psychotherapy has been chosen as a first field of application for Tele-Board MED research and interventions. This chapter shares insights and findings from empathizing with users, defining a point of view, ideating, and testing prototypes.

David Sirkin, Brian Mok, Sonia Baltodano, Dirk Rothenbücher, Srinath Sibi, David Miller, Jamy Li, Nikolas Martelaro, Nikhil Gowda, and Wendy Ju have developed a generative, improvisational and experimental approach to the design of expressive everyday objects, such as mechanical ottomans, emotive dresser drawers and roving trash barrels. They have found that the embodied design improvisation methodology has also been effective in designing the behaviors and interfaces of another kind of robot: the autonomous vehicle. "**Embodied Design Improvisation for Autonomous Vehicles**" describes their application of this design approach in developing and deploying three studies of autonomous vehicle interfaces and behaviors. Each study suggests design principles to guide further development.

2.5 Part III: Prototyping

Can anyone make a smart device? Affordable sensors, actuators, and novice microcomputer toolkits are the building blocks of the field we refer to as Creative Computing. With the growing maker movement, more tools are becoming available to novices, but there is little research into the usability evaluation of these toolkits. In "**Can Anyone Make a Smart Device?: Evaluating the Usability of a Prototyping Toolkit for Creative Computing**" Joel Sadler, Lauren Aquino Shluzas, Paulo Blikstein, and Sakti Srivastava discuss the importance of closing the gap between idea and prototype, the need for systematically evaluating the usability of novice toolkits, and a strategy for doing so. This work aims to contribute to the idea of "making simple things simple, and complex things possible," with prototyping toolkits of the future.

In "**Making Examples Tangible: Tool Building for Program Comprehension**" Marcel Taeumel and Robert Hirschfeld investigate prototyping in software engineering. Existing tools for program exploration are tailored to general programming language concepts instead of domain-specific characteristics and programmers' present system knowledge. In this chapter, the authors motivate the need for adapting the programming tools in use when navigating, viewing, and collecting examples to increase tangibility—that is, the clarity of a concept or idea based on what can be experienced on screen. In this context they present their Vivide tool building environment.

Appealing user interfaces and excellent usability are the keys to successful software products and services. However, great usability and user experience are not easy to develop because traditionally system engineers design solutions without involving end users. At the same time, current research suggests the involvement of end users in software development and the constant incorporation of testing and feedback to provide high-quality software and satisfying usability. In "**Case Studies on End-User Engagement and Prototyping During Software Development—An Overview of Current Practices in the IT Industry**" Franziska Häger, Sebastian Meyer, and Matthias Uflacker provide a look into the development process of three major software companies and present an overview of their current practices concerning end-user involvement and prototyping.

2.6 Part IV: Developing Design Thinking Teaching and Coaching Tools and Approaches

Design thinking has arguably become a state-of-the-art innovation methodology leading to an increasing demand for design thinking education. In "**Design Thinking At Scale: A Report on Best Practices of Online Courses**" Mana Taheri, Thomas Unterholzer, and Christoph Meinel aim to answer the question of whether and how design thinking can be taught in a form of Massive Open Online Courses (MOOCs), which deliver the promise of scalable teaching. In this chapter the authors discuss the opportunities as well as challenges in teaching design thinking in a MOOC environment. They propose recommendations for course designers, report on results of interviews with course instructors of the Stanford d.school on challenges and potentials of a digital design thinking learning environment, and present the path of future research.

When students work with peers, they learn more actively, build richer knowledge structures, and connect material to their lives. However, not every peer learning experience online sees successful adoption. In "**Designing Scalable and Sustainable Peer Interactions Online**" Chinmay Kulkarni, Yasmine Kotturi, Michael S. Bernstein, and Scott Klemmer first introduce PeerStudio, an assessment platform that leverages the large number of students' peers in online classes to enable rapid feedback on in-progress work. They then articulate and address three adoption and implementation challenges for peer learning platforms such as PeerStudio. They illustrate these challenges through their study of 8500 students' usage of PeerStudio and another peer learning platform: Talkabout. This research demonstrates how large classes can leverage their scale to encourage mastery through rapid feedback and revision, and suggests "secret ingredients" on making such peer interactions sustainable at scale.

Multidisciplinary teamwork is a key requirement in the design thinking approach to innovation. Previous research has shown that team coaching is an effective way to improve team performance. However, the tools currently available for effective team coaching are limited to heuristics derived from either experienced design thinking professionals or clinical psychology practitioners. The research of Neeraj Sonalkar, Ade Mabogunje, Halsey Hoster and Bernard Roth aims to improve this situation by providing design thinking managers, coaches, and instructors with a reliable instrument for measuring design team performance. In "**Developing Instrumentation for Design Thinking Team Performance**" they present the underlying methodology for instrument design. The development of a specific diagnostic instrument, based on a visual notation called the Interaction Dynamics Notation, is explained in terms of both the workflow of data through the instrument and the exploratory studies conducted to design the instrument user interface.

Additionally, in "**Stethoscopy for Design Teams: Instruments for the Exploration of Design Conversations**" Axel Menning, Andrea Scheer and Claudia Nicolai introduce two complementary instruments, the Knowledge Handling

Notation (KHN) and the Topic Markup Scheme (TMS). These instruments identify and analyze content-related and conversational patterns in team interactions. Both will be introduced and applied to a design conversation in an innovation team. TMS describes move-to-move coherence and global coherence. KHN describes on the move-to-move level how innovation teams generate and share knowledge. The output of both instruments, in the form of strings of symbols, can be used for sequence analysis and pattern detection of team dynamics. Together, the outcomes nurture the understanding of knowledge creation in and through design conversations in innovation teams.

The ability to produce novel yet appropriate (or useful) outcomes is broadly defined as creativity. So far, however, several methodological issues have restricted researchers in uncovering the brain basis for creativity and previous neuroimaging studies have for the most part produced varied findings, with little overlap. To partly mitigate some of these issues, the authors of the last chapter have recently developed a novel game-like and creativity-conducive neuroimaging paradigm that was employed to assess neural correlates of spontaneous improvisation and figural creativity in healthy adults. In "**Developing Novel Neuroimaging Paradigm to Assess Neural Correlates of Improvisation and Creative Thinking Using fMRI**" Manish Saggar, Lindsay C. Chromik, Adam Royalty, Grace Hawthorne, and Allan L. Reiss provide a brief overview of the current state of neuroscience research focused on creativity. They also provide insights regarding their experimental design, challenges faced during prototyping as well as a summary of their results. Lastly, building upon their novel paradigm, they provide pointers to future work for assessing neural correlates of creative capacity enhancement and team creativity.

2.7 Outlook

The publication at hand is the seventh of a series about Design Thinking Research and continues to share the findings from our HPI-Stanford Design Thinking Research Program with the public. We are pleased about the extension of this program for another 6 years that allows us to continue our work for a better understanding of design thinking, to investigate and develop tools and methods. As always, we are delighted to share and discuss our findings—not just with this and previous books but also through many other channels that have been established over the past years.

A very recent platform is thisisdesignthinking.net, a website that was launched by a project team of the Design Thinking Research Program. With the design thinking-related case studies and interviews presented there, enriched with scientific explanations, the researchers from the Hasso Plattner Institute provide insights from years of studying the application of design thinking in practice. This website is intended to serve as a pool of shared experiences from practitioners, scientists and coaches. It thereby presents the manifold perspectives on design thinking and meets the demand from practitioners, managers, entrepreneurs, and employees for more information about the application of this innovation approach. In the following chapter, the authors introduce this project and invite you to share your experiences on the website as well.

Furthermore, the Electronic Colloquium on Design Thinking Research (ECDTR, http://ecdtr.hpi.de) is an online forum using electronic media for scientific communication and discussions in the design thinking research community. It is ideal for the rapid and widespread exchange of ideas, methods, and results in design thinking research and welcomes papers, short articles and surveys.

To learn more about our Design Thinking Research Program we invite you to visit our website www.hpi.de/dtrp. It presents the latest information on past and present research projects, activities, publications, and community members.

We thank all authors for their contributions to this publication. Special thanks go to Dr. Sharon Nemeth for reviewing and copyediting them, as well as to Claudia Koch for preparing and coordinating the publication. Above all, we are grateful to Hasso Plattner for his constant support for our research and the extension of the Design Thinking Research Program for another 6 years. Over the past research years, meaningful knowledge about design thinking has been gained, yet so many more questions still remain unanswered. The field of research is broad, the demand for further insights both from academia and practice is rising. We are looking forward to continuing our work, expanding our knowledge about design thinking and sharing it with the public. This is our contribution to drive innovation in companies and society. We would be pleased to engage in dialogue with our readers for further discussions about your ideas, experiences, insights and questions via one of the above mentioned channels.

References

Altringer B (2013) A new model for innovation in big companies. Harvard Business Review, 19 Nov 2013

Dorst K (2015) Frame innovation: create new thinking by design. MIT Press, Cambridge, MA

Thisisdesignthinking.net: A Storytelling-Project

Eva Köppen, Jan Schmiedgen, Holger Rhinow, and Christoph Meinel

Abstract ThisIsDesignThinking.net was started in 2015 and is the first website dedicated to examine design thinking adoption in organisations. The blog is managed by an international network of editors-in-chief who contribute articles or advice authors.

> When does design thinking fail? What is the "right way" of implementing design thinking? Where can I find stories about how design thinking has been introduced in organizations, which are similar to the one I'm working in? In what ways can design thinking be meaningful to my organization?

These are some typical questions that are asked by practitioners, managers and employees, but also by social entrepreneurs and administrative staff within the public sector. This is why the "Impact by Design Thinking" project team (Eva Köppen, Holger Rhinow, Jan Schmiedgen, and Prof. Dr. Christoph Meinel) launched the website *thisisdesignthinking.net* in February 2015. *Thisisdesignthinking.net* showcases stories from companies working with design thinking. It also publishes interviews with experts and practitioners. Many years of research and insights on design thinking in organizations has led us to the idea of sharing our knowledge by way of this format.

1 Background

In 2015 we finished our study "Parts without a Whole" (Schmiedgen et al. 2015), an examination of design thinking adoptions in organizations. The idea of creating a website first emerged as a side project.

Every qualitative researcher knows the following situation: Conducting many qualitative expert interviews we realized that within the study we could rarely make accessible the vast amount of insights and «leftover knowledge» that we gained in our research. Only a fraction could be used for developing concepts for our study.

E. Köppen (✉) • J. Schmiedgen • H. Rhinow • C. Meinel
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: eva.koeppen@googlemail.com

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_2

Additionally, we have to admit that our scientific papers are rarely read by practitioners. From what we know practitioners are primarily interested in the anecdotal story chunks of other organizations. But unfortunately it is oftentimes exactly these side stories that are neglected in our academic publications. They tend to end up as defunct interview transcripts in our filing cabinets.

This is why we started our experiment. After getting our interviewees' permission to use their material, we compiled design thinking stories from the leftover information we had. This is how the first design thinking-related case studies and interviews on our website came into being. They are intended to demonstrate the wide range of perspectives on design thinking.

2 Continua of Design Thinking Practices

Our research perspective leads us to the conviction that there are multiple ways of practicing design thinking—none of them are either "right" or "wrong." Adopting a pragmatic point of view, we look for ways of integrating design thinking that either work for an organization or not. So instead of promoting the "right way" to practice design thinking, the website describes continua of organizational practices that are all outcomes of design thinking. These either make sense in their particular situation or have failed.

Because—more specifically—there is no explicit answer to the question of what design thinking is. As studies revealed, practitioners as well as academics have multiple interpretations in mind when it comes to defining design thinking (Schmiedgen et al. 2015). From a philosophy to a toolbox, from an innovation technique to an instrument for employee engagement—the range is so great that it would equal an inadmissible simplification to give a clear-cut definition.

This might be the reason why Kees Dorst, PhD. Professor at the University of Technology Sydney, Australia, demands from design research the articulation of "the kinds of design thinking and the ways they can be applied". With the cases and interviews demonstrated on this website, we hope to take further steps in this direction.

The website draws a colorful picture of the multiple design thinking activities that are going on today. It serves the research community as well as design thinking coaches, practitioners and students.

3 Who Is the Reader of Our Website?

The dynamics in the field of design thinking are breathtaking. The inflation of publications may soon become overwhelming for someone in search of orientation. *Thisisdesignthinking.net* offers an easy, accessible overview on current developments. Furthermore, such a pool of examples, enriched with scientific explanations, may help to cool down the oftentimes heated debate within the design thinking community and among practitioners. Such an overview helps localize all the existing kinds of design thinking with its pros and cons.

Practitioners searching for advice regarding design thinking get the chance to meet their "corporate twin" in organizations with similar preconditions, who may also function as role models. For educators, the website serves as a source for refreshing their exemplary materials, explanatory models and perspectives on current problems in design thinking practice.

4 The Future of Thisisdesignthinking.net

Do you of know an interesting design thinking story that is worth telling the world? If so, please get in touch with us. Prospective authors are experienced students, young professionals, PhD candidates, etc., who volunteer to write an article. In turn, they receive attention (our newsletter goes out to nearly 1500 persons) and acquire valuable contacts in the industries they are interested in writing about. Not only do the chosen organizations receive free "PR," but the people who are responsible for the design thinking initiatives are mentioned publicly.

Please contact us at: thisisdesignthinking@hpi.de

Reference

Schmiedgen J, Rhinow H, Köppen E, Meinel C (2015) Parts without a whole?—the current state of design thinking practice in organizations (Study Report No. 97) (p 144). Potsdam: Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam. Retrieved from http://thisisdesignthinking.net/why-thissite/the-study/

Part I
Design Thinking in Practice

Colliding Influences

When Self-Organizing Teams Encounter Strategic Objectives and Established Routines

Holger Rhinow and Christoph Meinel

Abstract This article illustrates findings from a case study on the impact of design thinking within a large organization. As teams apply design thinking as a framework for product discovery and development, there is an increased focus on the user as a source of inspiration as well as self-organizing teamwork. This phenomenon contrast to the result of other frameworks implemented in the organization (e.g., Waterfall and Scrum). These new influential factors are to some extent seemingly in collision with other existing influential factors, such as established routines in project management and a corporate strategy. Contrary objectives lead to paradoxical situations within teams and between teams and their managers. It appears that such situations can only be partially resolved by stakeholders. This case study empirically clarifies the impression from previous research that the integration of design thinking appears to be a managerial challenge yet to be mastered.

1 Introduction

The data in this article stem from a 6-month case study on the impact of design thinking in multiple product and service development projects within a large software company. It is one of the larger examples of an organization that embedded design thinking as a management model to (re)organize its development efforts.

Design thinking as a term (Rowe 1987) has been around for almost three decades, while its origins date back even further to discourses in cybernetics (Simon 1999) and social planning (Rittel and Webber 1973). However, the notion of design thinking as a management model has only been discussed for just over a decade (Buchanan 2004; Lindberg et al. 2010; Carlgren et al. 2011). The management discourse is particularly linked with the idea of design thinking as a model for

H. Rhinow (✉) • C. Meinel
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: holger.rhinow@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_3

enabling innovation. In its detailed form, the discourse varies from researchers who consider design thinking a resource (e.g. Brown and Watt 2010) to researchers who consider state design thinking a condition that can be nurtured through resources (e.g. Martin 2009; Bucolo and Matthews 2010)—with nuances in between. Nonetheless a few aspects are repeatedly mentioned within the discourse that seem to be ingredients of a design thinking experience, for example the relevance of a user-focus (Brown 2008), prototyping (Brown and Watt 2010) and self-organizing teamwork (Bucolo and Matthews 2010) around complex issues.

In recent years, the notion of design thinking as a management model has gained widespread attention in business contexts, in particular in organizations that reorganize their product and service development efforts (Clark and Smith 2008). There is empirical evidence that profit and non-profit organizations of all sizes and in almost all domains have adapted design thinking as a model for various purposes, most likely research and development. A recent study (Schmiedgen et al. 2015) shows that while most organizations have not yet achieved significant innovations in product development, they are more likely to improve their innovation culture within teams and units (71 %) and make their existing innovation processes more efficient (69 %). There seems to be a gap between the notion of design thinking as it is described in management discourse and the results that have yet been achieved in organizations. Interestingly, only a minority of respondents in the study blamed design thinking for any shortcomings in their organizations, but instead the management of design thinking or the organizational setup itself respectively (Schmiedgen et al. 2015, p. 106). As one interviewee stated:

> [Management] won't impose any kind of structure that's needed and that's the problem [. . .]. This is the core reason why rolling out design thinking [. . .] is a problem here: management cannot set a mandate. (Anonymous Interviewee from Schmiedgen et al. 2015, p. 111)

As the study focuses on giving a descriptive overview of the application of design thinking in organizations, it does not further elaborate on what form such a mandate would take/how such a mandate might look. It remains unclear how management seemingly fails to set up a structure that could potentially trigger innovation to a larger extent.

The following insights stem from a case study that aims to uncover the role of management in an organization that integrated design thinking in their product development. The results may give answers to the gap mentioned as they indicate that, through design thinking, new factors have become more influential. These new factors seem, to some extent, to contrast to other existing influential factors in the organization's product development.

2 Case Study

Data in this article stem from a case study of a large software company that extensively integrated design thinking in its product development in over 40 global projects. Most projects began in the spring of 2012 and ended in the fall of 2012 (Rhinow and Meinel 2014).

The primary author and another researcher from the same research program conducted 54 semi-standardized interviews with 44 interviewees of which 37 were directly or indirectly involved in one of the over 40 projects mentioned above. Nine out of the 36 stakeholders were interviewed twice, shortly before their projects started and shortly after their projects were completed/finished. One interviewee, a project manager was interviewed three times. The interviews lasted from 45 to 120 min (Table 1).

Interviews were conducted with external stakeholders (ES), who are either design thinking experts not affiliated with the particular organization or managers from the organization who are not involved in any of the projects. Team coaches (TC) are external and internal design thinking coaches who worked with at least one of the teams over the course of their project. Team members (TM) are those stakeholders who were constantly involved in the teamwork and did not have a designated leading role (e.g. developers). Project managers (PM) had a designated responsibility for at least one of the projects they set up. As part of the middle management, project managers were also responsible for communicating expectations and results from teams to strategic management and vice versa. Oftentimes project managers were involved in the teamwork activities themselves. Initiative managers (IM) are stakeholders who organized the setup of the initiative that comprised all of the design thinking projects within the organization. There were also Line managers (LM) who were strategically responsible for a collection of different projects within the initiative.

Table 1 Number of interviews per stakeholders

Stakeholders	Number of interviews
Externals (ES)	7
Team coaches (TC)	6
Team members (TM)	18
Project managers (PM)	19
Initiative managers (IM)	3
Line managers (LM)	1
Total	54

3 Data Analysis

The qualitative process of conducting interviews was based on the Grounded Theory Approach (Charmaz 2006; Glaser and Strauss 2009), whereby new topics emerge as relevant for the initial research question. These topics deal with the comparison of design thinking and further development models, such as Scrum and Waterfall. The notion of project management and strategy became significantly more important throughout the research process.

This case study will be published as a doctoral thesis and comprise a set of so-called aggregated research dimensions (Gioia et al. 2013) focusing on the notion of design thinking as a management model within the organization. This article presents a selection of two of the aggregated dimensions:

1. The notion of self-organized teamwork and the user as a source of inspiration as two factors in development projects that become increasingly influential in the presence of design thinking.
2. The collision of the above-mentioned influential factors with two established influential factors in product development: established routines of project management and a corporate portfolio strategy.

The authors applied the coding scheme of Gioia et al. (2013) to first code empirical data as 1st order concepts that can be aggregated and paraphrased into 2nd order themes. The interplay between various 2nd order themes defines the aggregated dimensions mentioned above. The coding scheme below illustrates examples of the coding structure for the two aggregated dimensions discussed in this article.

The empirically grounded and rigorous approach (Gioia et al. 2013) allows for the development of a grounded theory around the notion of design thinking as an enabler for the collision of influential factors in product development. This article will illustrate selected results of an emerging theory, which will be laid out in detail in the prospective doctoral thesis by the primary author (Table 2).

Table 2 Coding scheme selection

1st order concepts	2nd order themes	Aggregated dimensions
A valid concept for democratic teamwork	Increasing self-organization of teamwork	Increasing influential factors
Increased need for expertise and focus in teamwork		
Positive user feedback	Increasing user inspiration	
Demotivating missteps		
Teamwork focus	Self-organizing teamwork vs. project management	Collisions of influential factors
Preparation conflicts		
Cannibalizing projects	User inspiration vs. portfolio strategy	
Technology strategy		

4 Increasing Influential Factors

Among other factors, two stood out that shaped the experience of many of the involved team members, project managers, and team coaches.

First, within the projects the teamwork became increasingly self-organizing. This led to new opportunities and challenges for teams and management alike. One important indicator seems to be the fact that over the course of the projects, teams become aware that they have a lack of expertise on topics that emerge within the teamwork. These topics were oftentimes not anticipated by project managers and team members at the beginning of the projects.

Second, the methodological confrontation with users led to inspiring situations in many teams that would change the direction of their projects. The interviewed stakeholders compared their experiences to those of previous development projects within the same company.

4.1 Self-Organizing Teamwork

4.1.1 A Valid Concept for Democratic Teamwork

Several team participants described their project experience to be enriching, fun and a "valid parallel concept" (TM#8 September 2012) in addition to their established teamwork setups (e.g. in Scrum projects).[1] Even though aspects of teamwork in design thinking may look playful from the outside, participants consider the creative interactions to be "hard work" that they first learned to appreciate over the course of their project (TM#9 September 2012) (Table 3).

While several teams worked under the premise of open-ended results, their members stated a higher need for discussions about outcomes and opinions, resulting in "democratic" decision-making (PM#3 September 2012). This is a new element compared to previous projects in which project managers tended to have the final say about the content of projects.

Compared to the teamwork in Scrum projects, one initiative manager observed that "the dividing lines between team members' professions" started to blur (IM#2 September 2012). A project manager made similar observations and concluded that this notion had become a challenge for teamwork, as it demands a new, not yet definable way of thinking by all participants who are used to operating under specific roles and responsibilities (PM#18 May 2012).

[1] The comparison of design thinking and Scrum in particular, as it is perceived by the same stakeholders, is the subject of a previous article in this series (Rhinow and Meinel 2014).

Table 3 Increasing self-organization of teamwork

1st order concepts	Representative data
A valid concept for democratic teamwork	A lot of the things were new and super exciting. I can tell that like a lot of other colleagues, I learned a lot,. There were lows and highs [. . .] and things were fun in between, as well. [. . .] This change of perspective is at least a valid parallel concept to me. (TM#8 September 2012) A lot of things were playful from the outside, which felt a bit like a children's game for someone who normally works scientifically. In fact, it was hard work and many of us didn't even notice it at the time. Within our team we realized it now. Each one of us came to appreciate that creative work is also hard work—maybe even harder than everything else [. . .] All of us were drained in the evening [after a day of design thinking—editor's note]. (TM#9 September 2012) He [a colleague—editor's note] has a different opinion than the rest of the team, which is actually great. Thus a discussion developed or the others would say 'Wait a minute, this discussion is now over, We are four against one and we're doing it our way.' Things were done in a completely democratic and fair way. As I said, it was a relaxed atmosphere. (PM#3 September 2012)
Increased need for expertise and focus in teamwork	In the beginning we could not have said whether we would need someone [. . .] from automotive or someone from telecommunications [. . .] We started generic and [. . .] now we are at a point where we can say that it needs professional expertise so that you can further develop it. (TM#8 September 2012) It was a bit challenging because they [the team—editor's note] have worked on three different challenges, three situations. [. . .] They did not want to restrict themselves [. . .], which I think led to confusion. (TC#4 November 2012)

4.1.2 Increased Need for Expertise and Focus in Teamwork

Most stakeholders experienced design thinking to be the more open-ended and ambiguous framework for product development when compared to Scrum or Waterfall. Projects seemed to float in directions that were not completely foreseeable. In some teams, even the target industry domain was left open for team discussions and iterations, which felt unusual to the team members who were involved. As a result, a few teams developed in directions where they did not have a particular expertise. One team member stated that he could not have foreseen the direction the team would take as they began taking into account various industries and various technological options as a basis for their research. As it turned out, their solutions would gravitate both toward the automotive and the telecommunication market, neither of which any of his colleagues in the team were acquainted with. At this point "professional expertise" would have been needed to develop the project further (TM#8 September 2012).

As one project manager said, the need to integrate experts became apparent only later in the process; in fact, too late to react from a resource perspective. Different team members can only speculate in hindsight why areas crystallized so late in the process. With the chance of defining the goals of the projects themselves, the team missed the chance “to restrict themselves” early on (TC#4 November 2012) as they have had high expectations to generate solutions for “three different challenges” simultaneously as one coach observed.

4.2 User Inspiration

4.2.1 Positive User Feedback

A majority of interviewees valued the chance to get in contact with users as part of a user research phase. As one project manager stated, her team gave her “consistently positive” feedback (PM#3 May 2012). Another team member remembers customer visits as being “interesting and motivating” for her own work (TM#13 September 2012). Team members recognized the new approach to be an opportunity for them to get to know users and their needs in an unfiltered manner, something they had “never done before” (PM#14 September 2012) (Table 4).

Some teams were surprised that their initial assumptions and perceptions about users and their needs turned out to be false during these interactions. For example, one team aimed to re-design a ticketing feature for technical issues for internally used software solutions. A previously conducted survey showed that many employees were dissatisfied with the actual service. The project manager believed that employees would prefer a community-based approach in which employees could help each other directly in dealing with technical issues instead of writing tickets to an anonymous service unit. A common approach in other project set-ups would be to define a backlog around the idea and have a team execute the tasks. In the actual project, the team started to engage in interactions with users to find out reasons for their dissatisfaction with the current feature, as well as their needs in general. It became apparent, that most of them were “not interested at all in a community-based solution” (PM#3 September 2012) in this context, which was highly surprising to the project manager. As a matter of fact, the seemingly anonymous ticketing feature provided a feeling of security. Employees felt that they were able to address someone who was in fact responsible for and capable of dealing with their tickets. Even though they were dissatisfied with the current ticketing solution, none of them were unhappy with “writing tickets” (PM#3 September 2012). On the contrary, most employees responded negatively to the idea of a community that would support each other, as these were probably people, that wanted to distinguish themselves and one could not be sure whether their answers are correct (PM#3 September 2012). It turned out that that ticketing system itself was regarded as a positive idea but the actual solution was seen as “too slow” (PM#3 September 2012). The team dismissed their initial idea of a

Table 4 Increasing user inspiration

1st order concepts	Representative data
Positive user feedback	The feedback was consistently positive. Comments included: 'it was great. More interviews, we want to know more about our end-users.' Throughout the team. There was one guy, he was so happy, he would love to do just interviews from now on. (PM#3 May 2012) It was a positive experience, very interesting to get in contact with users, end-users. It was interesting and motivating. (TM#13 September 2012) [In previous projects—editor's note] you may have given a sales pitch in front of your customer and they might have given you feedback on what they dislike [. . .] but going to them and wanting to get to know their problems, that is something we have never done before. I found that to be very, very interesting, very good. (PM#14 September 2012) And then we were surprised, everyone said 'okay, this is something we haven't thought of' [. . .] We were surprised that people were not interested at all in a community-based solution [. . .] and everyone just wanted to write tickets. (PM#3 September 2012)
Demotivating missteps	One of the key moments we had was the introduction of our very first prototype to our customers [. . .] their feedback was that they were actually not interested at all in the topic [. . .] that was very surprising for the team, partially demotivating for the team, as well since we did not know how to move on from there. (PM#2 September 2012) After it went down the wrong path [with the first prototype—editor's note]—based on customer's feedback, I strongly demanded that we now have to consider on what to focus [. . .] as did not want to see another misstep. (PM#2 September 2012) We have worked on something for 2 weeks, in the end we have tested it with the customer and on paper and discovered that we had completely misunderstood each other [. . .] We wasted 2 weeks of time. (TC#3 December 2011)

community-based solution and began to redesign the service process around the ticketing solution.

4.2.2 Demotivating Missteps

Several teams experienced their users as ambiguous, unable to always state their needs precisely. Therefore, many interview results remain provisionally valid until the teams actually start to prototype solutions they can test. As these tests were somewhat open for feedback on the core idea instead of its usability, teams often experienced surprising results that would question their previous assumptions about their users.

One project manager remembers the testing of his team's first prototype as a "key moment" (PM#2 September 2012). The team built a prototype around the idea of a software solution that would automatically delete outdated data. This is data that would otherwise be stored on servers until someone manually deleted it. First interviews affirmed the team's assumption that specifically IT managers in large companies would highly value the idea. The team prototyped the solution as a series of interface sketches combined with information about the algorithm that would demonstrate how the automated deleting would proceed. By showing their prototype to actual users, the team wanted to receive open feedback. In particular, they wanted to understand how much people could trust the idea of automatically deleting company data. To the project manager's surprise, most users were not interested in the general idea of automatically deleting data, as they considered other issues more pressing (PM#2 September 2012). A team member remembers that the initial interviews already indicated that users had different priorities, but she believed that they might have "not listened 100 %" (PM#2 September 2012) to what these were at that time. The testing results were "partially demotivating" for the team and led to the decision to start all over again. During the second iteration, the team designed a prototype that was then well received by the same users. This was, in particular, a relief for the project manager. Even though he acknowledged the fact that the initial testing saved the team from developing a hardly desired solution, the project manager still wanted to avoid "another misstep" (PM#2 September 2012) for his team.

The notion of a failed prototype remains ambiguous especially to some of the interviewed stakeholders. While some understand it as an opportunity to learn, others perceive it as a misstep or a waste of time to be avoided (TC#3 December 2011).

5 Collisions of Influential Factors

As user inspirations and self-organizing teamwork begin to shape the project experiences of stakeholders, other influential factors seem to collide, to some extent, with these emerging factors. The notion of an increasing self-organization of teamwork becomes a challenge in particular for project managers. This challenge occurs when project managers feel both a need to maintain a shared focus within the teamwork and a need to individually prepare preliminary content for further elaboration by the team. While these managerial actions are common in other teamwork set-ups, they now seem to lead to critical, if not paradoxical, situations within the team.

5.1 Self-Organizing Teamwork vs. Project Management

5.1.1 Teamwork Focus

Many project managers, but also team members, regarded the open-ended approach to be problematic in hindsight. It turned out to be more difficult than expected to find a focus for their teamwork while, as they ran out of time. One team member felt

Table 5 Self-organizing teamwork vs. project management

1st order concepts	Representative data
Teamwork focus	We realized, we have to find and end point, otherwise we are blathering. We are trying to solve the problem of world hunger. (TM#8 September 2012) We did too many things in parallel for too long [. . .] I think we should have acted more focused. (TM#8 September 2012) It felt like a hodgepodge of insights. (PM#3 September 2012) Maybe it would have been more important to focus early on or earlier and to say 'okay, this is the direction.' (PM#3 September 2012)
Preparation conflicts	I understand the frustration of the [. . .] upper management for not understanding why things take so long. [. . .] Because once you have too many people that also makes you very slow. Five thousand developers is a lot of developers to coordinate. (PM#11 March 2012) They already built a mockup [. . .] I did not see it but it was there [. . .] and it was real in their [the project manager's and developer's] heads. (TM#1 September 2012) If you already know, what the customer wants, then why are we doing this [kind of work], why don't you look for a designer who just builds it? (TM#1 September 2012)

that his team was "trying to solve the problem of world hunger" without ever reaching a resolution (TM#8 September 2012). A member from another team missed the time when his team was able to strongly focus on one thing instead of doing so many things "in parallel for too long" (TM#8 September 2012) (Table 5).

Before the projects started, one project manager stated that she perceived design thinking to be a chance to think completely free (PM#3 May 2012). At the end of the project she then concluded that the open-ended approach turned out to be "really difficult" (PM#3 September 2012) for her and the team. The results felt like a "hodgepodge of insights" (PM#3 September 2012). The team was not able to pinpoint the user's problem in a way that would have allowed the designing of "a real scenario" (PM#3 September 2012), instead the team ended up with "seemingly hundreds of ideas without much of depth to them" (PM#3 September 2012).

Based on her experience, she now assumes that it is "more important to focus early on and say 'okay, this is the direction'" (PM#3 September 2012). For more than 8 weeks, the team struggled to decide what to focus on, instead they continued to keep options on the table and further add ideas. Only in hindsight did she realize that the problem they were tackling was "not such a big problem" as they had thought in the beginning (PM#3 September 2012).

Several project managers seemed to experience similar situations in which they felt the urge to focus the teamwork but perceived it to be challenging since the teamwork became more and more self-organized (PM#14 September 2012).

5.1.2 Preparation Conflicts

Several project managers showed ambitions to prepare content in advance as they have done in previous projects, as well.

One of the projects dealt with the idea of delivering a software solution to sales people that is mobile and that focuses on core features. The project manager decided to conduct most of the interviews with a colleague 3 months prior to the project and had already defined requirements from these interviews. This made sense for the project manager, and design thinking was perceived as another tool to speed up development processes that had become too slow in the past:

> I understand the frustration of the [. . .] upper management for not understanding why things take so long. [. . .] Because once you have too many people, it also makes you very slow. Five thousand developers is a lot of developers to coordinate (PM#11 March 2012).

For the project manager, previous development efforts had seemed like a "trauma because of the many years involved [. . .]. There were so many iterations and so much money was put into it" (PM#11 March 2012). Design thinking therefore seemed to be the answer to how to become more agile in development projects.

From the project manager's perspective, it seemed reasonable to begin the project with just one other colleague. After she and her colleague, a developer, defined all requirements and designed a first prototype, other team members joined the teamwork. During the first weeks of the project the team set up new interviews to validate the requirements but did not discuss the initial prototype. As one team member stated, she was aware of the fact that "they already built a mockup", however she "did not see it but it was there [. . .] and it was real in their [the project manager's and developer's—editor's note] heads" (TM#1 September 2012).

As the team conducted new interviews and took part in various brainstorming sessions, new ideas came to light on how to serve a sales person's needs with a mobile solution. As it turns out those new ideas differed significantly from the project manager's initial prototype. At some point in the teamwork the project manager mentioned that she is "well aware of what the customer wants" (TM#1 September 2012). It is, however, not what the team came up with. Two of the team members now openly addressed the knowledge gap between the project manager, the developer, and the rest of the team. Feeling disappointed, one team member told her project manager that if she already knew what the customer wanted then why were they "doing this [kind of work], why don't you look for a designer who just builds it?" (TM#1 September 2012). The project manager was initially convinced that the setup was reasonable in order to avoid another development project that was "missing a clear focus" and that would "support all the little deviations [. . .], which amounted to approximately 0.01 % of the usage" of a software solution (PM#11 March 2012). However, confronted with dissatisfaction within the team, the project manager agreed to re-examine the prepared prototype against the background of the team's shared insights. Over the next days, the team designed new mockups that looked "completely different than the initial mockups" (TM#1 September 2012).

5.2 User Inspiration vs. Corporate Portfolio Strategy

5.2.1 Cannibalizing Projects

From the perspectives of most interviewees, design thinking and Scrum can be regarded as complementary forms of teamwork that most likely will occur sequentially one after another. While teamwork under design thinking is expected to define a valid idea from a user's perspective, Scrum teamwork is expected to efficiently develop a defined idea (Table 6).

During the initiative one could observe that a few project managers set up parallel design thinking and Scrum projects. One project manager and her upper management believed that one could simultaneously define the idea for a software frontend with a design thinking team while a Scrum team would develop the backend infrastructure (PM#3 May 2012). In the beginning of both projects she thought of this constellation as a "perfect match" (PM#3 May 2012), since both parties could focus solely on two very different aspects: the requirements of end-users and aspects of efficiency or productivity. The project manager regarded herself as the "interface" between both teams as she updated both teams on current insights from the respective other team (PM#3 May 2012).

After both projects ended, the same project manager seemed skeptical about the setup. She realized that especially the ideas that came out of the design thinking team could not influence the development project as it "did not work out time-wise" (PM#3 May 2012). The end-users needs seemed to differ significantly from the initial assumptions; therefore the recommended changes for the infrastructure could not be addressed within the duration of the development project. In the end, the

Table 6 User inspiration vs. corporate portfolio strategy

1st order concepts	Representative data
Cannibalizing projects	I assume we will achieve a compromise in the end. One will not optimize one or the other to the fullest [. . .] we want to get to the perfect solution [. . .] but in the end [. . .] coming to a win-win situation for everyone (usually) means achieve something in the middle. (PM#3 May 2012) Once the idea is executed, which is to say it gets build, there is not much left for me to do, [you then have a] Scrum team that develops hardcore [. . .] that's when "the curtain drops." (TC#6 December 2011) The team needs a lot of lead-time, for organizing and for coordinating because coding new components isn't simply like pulling a rabbit out of a hat; if you do this you will cannibalize other teams. (TM#9 September 2012)
Technology strategy	It was important for us to say 'okay, we need a project that applies [the technology—editor's note] and on the other hand to be able to demonstrate that it works.' (PM#18 May 2012) We are of course not totally free to do what we want. (PM#18 May 2012) I realized that we "lose the wows" because we are a technology company. This means that we always try to reduce a problem to know aspects [such as—editor's note] existing technology. (TM#9 September 2012)

development turned out to be a "compromise" (PM#3 May 2012) with a few insights from design thinking to be addressed in the final solution.

One coach who supported several design thinking and development projects mentioned the limitations design thinking teams experience as soon as the development phase begins:

> Once the idea is executed, which is to say it gets build, there is not much left for me to do, [you then have a] Scrum team that develops hardcore [. . .] that is when the curtain drops (TC#6 December 2011).

At least two project managers experienced conflicts of interests between parallel design thinking and development projects. One project manager stated that the conflict is not necessarily a time issue but a scoping issue. From his point of view, many development projects aim to deliver a release update of an existing solution that would add several new features demanded by sales representatives. If the outcome of design thinking was a different set of features, then those stakeholders would

> go on the warpath and this would escalate to management, with the sales rep saying 'sorry, I need those [. . .] ten features. Without them, I cannot deliver'. There would be release stops, customer complaints and so on, and eventually the design thinking project would come to a halt. This is where theory meets reality (PM#13 April 2012).

One team member concludes that design thinking can only work within constraints that are set up before the projects start. Otherwise teams would just define new components, and these might "cannibalize" (TM#9 September 2012) the efforts of other teams. Management therefore needs to align these efforts by negotiating goals for different projects with various units.

5.2.2 Technology Strategy

Technology is a key resource for the development of innovations within the software company. Over 40 years, the company developed a number of technologies that became the foundation for numerous software solutions addressing multiple industries. In this regard technology has always played an important role for the strategic planning of portfolio development. Oftentimes managers were asking themselves what new opportunities the technology would offer to implement instead of understanding what use cases were actually needed from a users' perspective. While design thinking encouraged teams and managers to consider the user's perspective as a major driver for new developments, the notion of technology itself as a driver for new developments remained also present. This led to conflicts in a few teams that aimed to openly imagine solutions solely based on users' needs while their project managers stressed the importance of existing technologies.

One project manager explicitly made clear to her team that the existing technology was expected to become an integral part of the upcoming solution. This is because her management expected a solution that "applies [the technology—editor's note] and that would demonstrate that it works" (PM#18 May 2012). During the project, she realized the impact of the technological constraints in admitting, "we are of course not totally free to do what we want" (PM#18 May 2012). Furthermore, she states that the company originally would come from a technological perspective which led to the idea that you could "solve a problem without knowing the problem, really" and that "this is how things are running inside the company" (PM#18 May 2012). Her team would later end up with a use case for a new technology that was as she expected it.

A team member from another project who also worked under the constraint of a given technology felt that "we lose the 'wows'" because of this technological focus. The team did not consequentially follow the problem based on the user research but instead shifted to a problem that could be addressed with the existing technology.

6 Conclusion

The illustrated case study provides empirical insights that show how design thinking as a management model in product development shaped the teamwork in ways that most team members and project managers have not experienced before. Confronting the user in order to gain empathy und understand unmet needs became a major source of inspiration for the teams. The teamwork itself was perceived significantly different as well. Teams began to self-organize and started to reframe the content of projects to a large extent. Some teams even redefined the scope of their projects to an extent that they realized they were lacking the necessary expertise to further design concrete solutions. However, many teams realized they could not deliver solutions that were as promising as they had hoped to do in the beginning.

While the project setups allowed for unforeseen new situations, established strategic and organizational routines around the teamwork itself remained present, such as the influence of a corporate portfolio strategy and routines of project management. As it turned out, these established factors remained significant and to some extent came into collision with the influential factors triggered by design thinking. It is not said that the company would be better off by neglecting their established routines in order to end up with more innovative solutions, however empirical results indicate that project results are less likely to become innovative once seemingly contradictory factors collide within the teamwork itself.

Previous research indicated the importance of management for setting up a design thinking landscape that would nurture innovation (Buchanan 2004). Now, empirical results further concretize those managerial challenges (e.g., as a challenge to coordinate contrary influential factors in product development). In this context, a design-thinking-oriented management will less likely play a role in a command-and-control structure. It will more likely become a horizontal interface between projects and business units that begin to discover new opportunities but that do not want give up on the possibility of synergies between each other.

References

Brown T (2008) Design thinking. Harvard Business Review, pp 84–92
Brown T, Watt J (2010) Design thinking for social innovation. Stanf Soc Innov Rev 8(1):28–35
Buchanan R (2004) Management and design: interaction pathways in organizational life. In: Boland RJ, Collopy F (eds) Managing as designing. Stanford Business Books, Stanford
Bucolo S, Matthews J (2010) Using a design led disruptive innovation approach to develop new services. In: Kobe C, Goller I (eds) Proceedings of the 11th international CINet conference: practicing innovation in the times of discontinuity. CINet, Zurich
Carlgren L, Elmquist M, Rauth I (2011) Implementing design thinking—an exploratory study of large companies using design thinking in innovation efforts. In: Proceedings of the 2011 Tsinghua-DMI international design management symposium HK in Hong Kong, pp 1–19
Charmaz K (2006) Constructing grounded theory. A practical guide through qualitative research. Sage, Thousand Oaks
Clark K, Smith R (2008) Unleashing the power of design thinking. Des Manag Rev 19(3):7–15
Gioia D, Corley K, Hamilton A (2013) Seeking qualitative rigor in inductive research: notes on the Gioia methodology. Organ Res Methods 16:15–31 (originally published online 24 July 2012)
Glaser B, Strauss A (2009) The discovery of grounded theory: strategies for qualitative research. Transaction Publishers, New Brunswick
Lindberg T, Noweski C, Meinel C (2010) Evolving discourses on design thinking: how design cognition inspires meta-disciplinary creative collaboration. Technoetic Arts 8(1):31–37
Martin RL (2009) The design of business: why design thinking is the next competitive advantage. Harvard Business Press, Boston
Rhinow H, Meinel C (2014) Design thinking: expectations from a management perspective. In: Plattner H, Meinel C, Leifer L (eds) Understanding innovation. Springer, Heidelberg
Rittel H, Webber M (1973) Dilemmas in a general theory of planning. Policy Sci 4(2):155–169
Rowe P (1987) Design thinking. MIT Press, Cambridge, MA
Schmiedgen J, Rhinow H, Köppen E, Meinel C (2015) Part without a whole? The current state of design thinking practice in organizations (study report no 97). Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam, Potsdam
Simon H (1999) The sciences of the artificial. MIT Press, Cambridge, MA

Mapping and Measuring Applications of Design Thinking in Organizations

Adam Royalty and Bernard Roth

Abstract Design Thinking is a methodology many organizations use to drive creative innovation. As this practice continues it is critical to develop tools that accurate describe how design thinking is being applied in teams and across an organization as a whole. This chapter introduces two tools in development to meet these goals. The first is an Ecology Mapping that portrays an organization's internal design thinking strategy. The second is a weekly "snapshot" of design thinking activities performed by industry teams working on creative projects.

1 Introduction

A number of measures that describe the impact of design thinking on an individual level have been created. They include psychological measures (Royalty et al. 2014), neuro-cognitive measures (Saggar et al. 2015), and performance tasks (Hawthorne et al. 2014). However, tools that capture the work of multiple individuals collaborating together within the bounds of an organization are needed. For the past 5 to 10 years dozens of major organizations have sent employees to multi-day design thinking trainings. In fact, a number of startups have formed in the past few years that specialize solely on providing design thinking trainings. As more individuals are trained, organizations have the ability to create teams, groups, and strategies that rely on a number of people having some level of design thinking expertise. And while individual measures are key to developing employee capacity, organizations must be able to describe their larger efforts in a way that helps them make informed iterations.

This chapter introduces two pilot studies aimed at mapping how design thinking is applied throughout an organization. The first is an Ecology Mapping of Design Thinking that describes the projects and people that are applying design thinking

A. Royalty (✉) • B. Roth
Hasso Plattner Institute of Design (d.school), Building 550, 416 Escondido Mall, Stanford, CA 94305-3086, USA
e-mail: aroyalty@stanford.edu

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_4

within an organization. In essence, this tool characterizes an organizations' internal design thinking strategy. The second study measures how teams practicing design thinking perform. The specific focus within the study is on empathy—a core component of design thinking and one that the subjects highly value. Together the two tools being developed and tested are an extension of individual design thinking measures. Eventually this will help organizations better assess how they deploy design thinking across teams and entire business units.

2 Study 1: Developing Ecology Maps of Design Thinking in Organizations

2.1 Background

Dozens of companies have made large commitments to design thinking (Schmiedgen et al. 2016) within the past 10 years. They do so for varying reasons depending on their innovation goals (Royalty et al. 2014). These commitments have led to a clear and growing need: organizations need a better understanding for how design thinking can be used on strategic and project levels. The need is highlighted by the ever increasing number of inter-organizational communities forming strictly to share design thinking practices. In year 1 of this study (2013–2014) we began studying one of these (Royalty and Roth in press). In the current year (2014–2015) we have connected with three more. Being a part of these communities has allowed us to capture the ways members of the communities apply design thinking. Beyond that, the questions community members ask each other reveal the real challenges they face.

To satisfy this need of understanding how design thinking is applied we seek to map out how each organization is applying design thinking, who in the organization is doing it, and why design thinking is their paradigm of choice. This responds to three main research questions:

> Within an organization, what are the projects, programs, and people that make up an ecology of design thinking?
> How do these ecologies change over time?
> What causes them to change?

Developing an accurate and robust framework that can map multiple companies has clear value to practitioners, as well as contributing, to organizational theory. Amabile's model of innovation in organizations (Amabile 1996a) provides a theoretical framework we have used to inform our map. In her cyclical model, work environment impacts team creativity, while team creativity drives innovation. The work environment is made up of three components: organizational motivation, management practices, and resources. Organizational motivation represents the strategic goals for innovation. Management practices capture how the leaders support the creative work of employees. Resources is a broader category that includes initiatives, human capital, and more. These environmental components provide a great lens with which to look at our organizations' ecologies. Therefore, any mapping framework needs to include them.

One might argue that the three components of team creativity should be included in the mapping. After all, teams are part of organization. But because we seek to understand the context within which teams and individuals work, we decided to use only the environmental components of Amabile's model. As this work continues, it may make sense to leverage the rest of the model.

Our work consisted of two main steps. The first step was to understand the key aspects of the ecologies that must be mapped. This allowed us to create the framework. The second step was to collect the relevant data in order to create an initial map of each organization. To do this we collected a mix of qualitative and quantitative data.

2.2 Methods

Data were collected on the use of design thinking from two groups. One of these was through the four communities of practice (each consisting of multiple organizations), and the other targeted interviews at seven organizations.

With regards to the four communities, the primary researcher, was a participant observer in convenings of the four separate communities (Communities A, B, C, and D). Our role in each community was to simply convey what we have learned and to capture what others shared. We avoided driving any agendas or advising organization how they should apply design thinking. The four communities are:

Community A: This group is made up of four companies that come together to teach each other's employees and share design thinking best practices. It was a focus of year 1. The group has existed for nearly 18 months and has about a dozen regular members. There are bi weekly phone calls to support continual collaboration. The community has convened in person twice and will do so again in early July. We have been a party to all but one phone call in the past year and attended every convening, and will attend in July as well. We capture meeting notes (for ourselves and the community) plus design the reflections used during the in person sessions.

Community B: A large technology and communication firm created a network of design thinking practitioners by training IT teams from client companies. They started 7 months ago and have teams from nearly 20 different organizations (including the host company). There are monthly phone calls that were preceded by two in person training sessions. We have participated in the majority of the phone calls and interviewed participants of the training session, though we were not in attendance.

Community C: Five organizations self-formed to connect for a regular video conference primarily around the topic of measuring design thinking. These sessions began about 4 months ago. There have been three meetings with a fourth scheduled for July. We joined the last session and will attend in July.

Community D: A large medical organization held a 2-day workshop in May connecting design thinking practitioners and researchers. There were representatives from 11 companies and 8 universities. The goal was to launch joint projects between industry and academia. The first of multiple follow up calls is scheduled for late June—which we plan to attend. Much of the data for the mappings and feedback on our framework was provided during the workshop.

To further our understanding of design thinking application, we conducted eight in depth interviews of practitioners from seven different organizations. All the organizations were members of a larger community. The interview subjects were selected based on their leadership role in the network (e.g. meeting attendance and role). They range from entry-level positions to senior leaders. We used the open ended interview protocol developed in the first year of this study (Glaser and Strauss 2009; Royalty and Roth in press). The interviews were open coded with four general categories emerging: people practicing design thinking, projects that use design thinking, programs that use design thinking, and unknowns. The difference between project and programs is mainly one of size. Projects typically involve one or two teams working to solve a specific business goal (e.g., how do we help elderly feel more financially secure). Programs are large and involve many more people (e.g., an incubator program for ten teams). Unknowns are the explicitly stated questions practitioners have for their colleagues from other organizations. We conducted follow up interviews to gather more details to feed our map.

2.3 Initial Results

Using interview and observational data from in depth interviews and community participation in tandem with Amabile's framework, we created an ecology mapping framework consisting of three components for each organization. These components correspond to the three parts of Amabile's work environment and are illustrated in Fig. 1.

The Innovation Target 2×2 (Fig. 1a) shows where the design thinking efforts fall relative to a general innovation framework. Incremental or breakthrough innovations that focus on cost savings or revenue generation. This relates to Amabile's organizational motivations. We plot known design thinking projects and programs that exist in any given year for each organization.

The Design Activities Diagram (Fig. 1b) captures how much of each activity an organization is doing. This relates to Amabile's resources. There are four distinct axes, resulting in a spider diagram of each organization. What will be important is the general shape of the resulting diagram. For this iteration we chose axes of: experts (number of), employees trained (percentage of total workforce), training (number of events per year), and projects (number of projects per year).

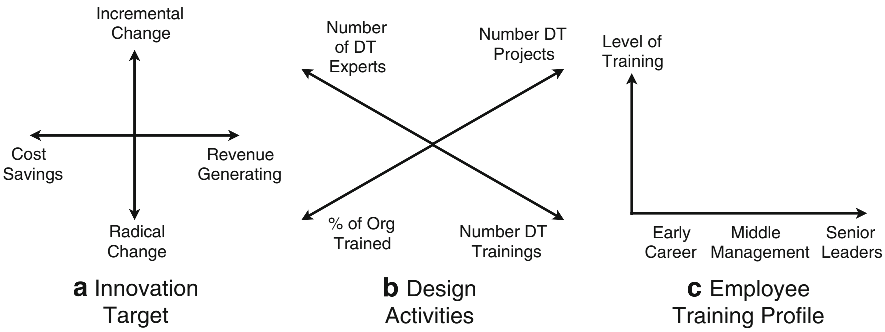


Fig. 1 Ecology mapping framework. (**a**) Reflecting management, (**b**) reflecting resources, (**c**) reflecting management motivation practices

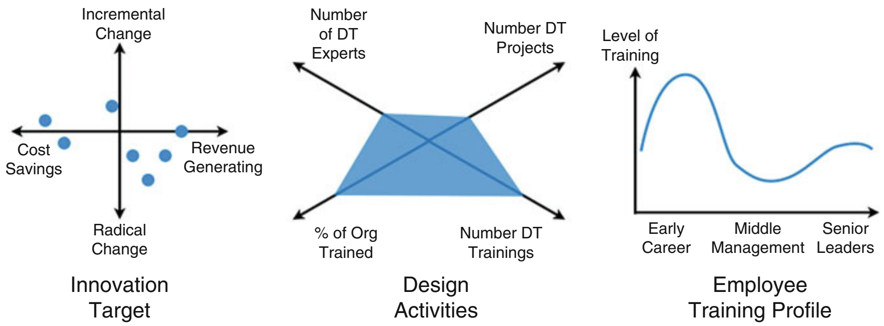


Fig. 2 Ecology map of a large high tech firm for 2014

Finally the Employee Training Profile represents the depth of design thinking capacity in a workforce and where that capacity is located along a leadership spectrum. This relates to Amabile's management practices. This chart captures the distribution of activity. Design activity is a combination of practicing, leading, and teaching designing thinking. The horizontal axis shows how much design activity exists in different leadership levels.

The initial mappings based on the data collected thus far are presented below.

The map in Fig. 2 shows a relative balance between trainings and project work with a slight bias towards training. There is a range of cost savings and revenue generating projects with more of a focus on radical change. The Employee Training Profile reveals what we hear from many organizations; design thinking capacity at the top and bottom, but not the middle.

This organization represented in Fig. 3 does not have as much design thinking investment as the one above in Fig. 2. You can see only four projects, and a number of trainings done by a few experts on a small part of the organization. This indicates that they are targeting design thinking on a few people. Notice that all the projects are focused on incremental change. This may be a result of the tight margins and low risk profile of the industry.

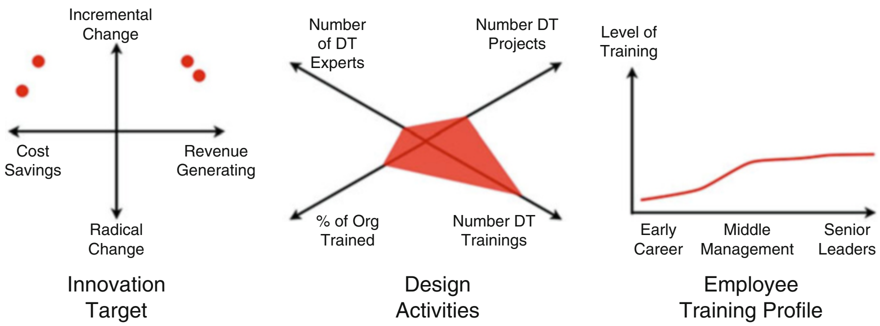


Fig. 3 Ecology map of a large transportation company for 2014

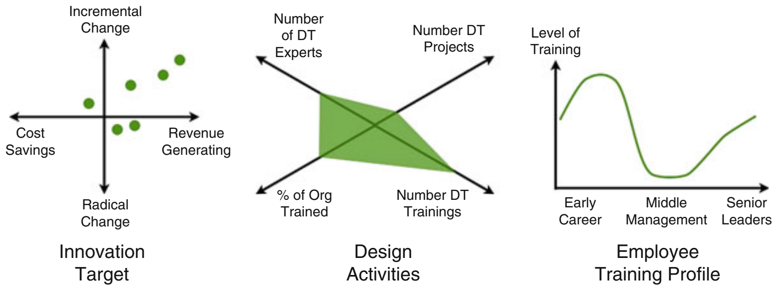


Fig. 4 Ecology map of a large financial services firm for 2013

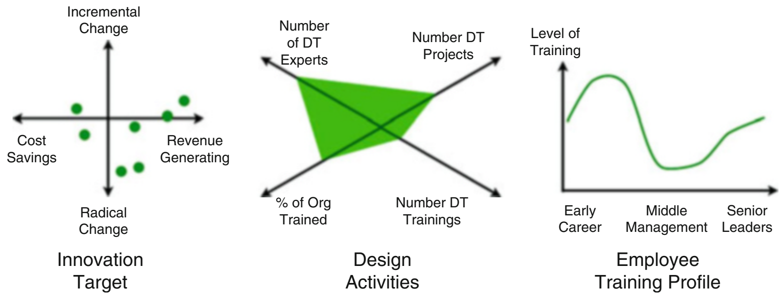


Fig. 5 Ecology map of a large financial services firm for 2014

Figures 4 and 5 illustrate a single company across 2 years. In 2013 the company heavily focused on trainings and projects leading to incremental change. The following year they greatly reduced the number of trainings and increased the number of design thinking experts and projects. The projects also tended to focus more on radical change. This reveals a strategic shift the company made. They initially wanted design thinking to be a cultural value of all the innovation teams. However, the pressure for business results in the form of new products meant that the design thinking team needed to shift gears. They moved their experts to support real teams and created an incubator program that invested in specific high potential projects rather than continue to teach as many people as possible.

3 Study 2: Measuring Team Behaviors and Outcomes

3.1 Background

Measuring is a separate but necessary complement to mapping. If the goal of mapping is to see what actions are happening and how they are being executed, the goal of measuring is to understand value of these actions. Measuring design thinking is difficult partly because it is a multifaceted paradigm. It is also contextually dependent (Martelaro et al. 2015). There is some existing work on measuring design thinking. Much of it stems from classroom or laboratory studies (Goldman et al. 2012; Sonalkar et al. 2014; Hawthorne et al. 2014). Measuring design thinking in organizational contexts is relatively unexplored. However, previous work on this project yielded a good starting place; principles for measuring in design thinking in real settings (Royalty and Roth in press). Namely measures should be easy to use and align with the organization's innovation goals.

Capturing any complex set of behaviors in context is a difficult task. Still, there have been studies that have successfully accomplished that task. Csikszentmihalyi captured subjects actions and "random" times by paging them and having them capture what they were doing (Csikszentmihalyi and Larson 1987). This contributed to his theory of flow. Another study by Amabile measured creative activities employees performed via a daily journal (Amabile et al. 2005).

Based on these previous methods we developed a simple measurement tool aimed at collecting individual design thinking behaviors exhibited at work. It is important to note that we defining behaviors as actions taken that support design thinking methods and mindsets. For example, talking with potential customers is a behavior that supports need finding. We intentionally chose not to capture techniques, like asking open-ended questions, because they are often subtle and difficult to detect. Also, we believe that the value of design thinking is in affecting behavior, not simply applying tools. Finally, our goal is to link behavior and outcomes. We want to show that people who work in way driven by design thinking generate more creative outcomes. This is the focus of our main hypothesis:

> Design thinking behaviors will be positively correlated with creative outcomes.

The first subjects were employees at a large North American healthcare management company that we will call Canyon Healthcare. The measures were included as part of the Canyon Healthcare Leadership Program (CLP). The leadership program is comprised of approximately 30 middle managers. These are current Canyon employees that have been identified as the company's next leaders. The program lasts 10 months, starting early May and going through February. Each participant is expected to work about 15 % of their time on CLP activities. The central challenge of the program is to use design thinking to tackle an ambiguous problem specifically outside their skill set, i.e. make the hospital discharge experience more delightful. The program kicks off with a design thinking training and features two additional design thinking sessions throughout the year. Participants work in teams of four and check in with a trained design thinking coach at least once a week. The project sponsors are senior Caynon leaders who review the outcomes at the end of the program.

CLP is a good setting to explore our measure for three reasons. The first is because all the teams have the same training and schedule. That means they will essentially be focusing on certain behaviors at certain times. The second is the 10-month time constraint. The projects have a fixed amount of time and the duration is enough to collect a sizable amount of data. Finally, the existing mechanism of critique by both coaches and senior leaders produces a relevant measure of creative output; that is the goal of the program. The major drawback is a lack of a control group. However, we can compare the creative output of all the teams and compare that to the amount of behaviors each reported.

3.2 Methods

The behavior measure we developed captures a weekly "snapshot" of activity. Every Thursday we send an email out to all CLP participants asking them to respond to a prompt. Each prompt has a numeric component and a short answer component. See Table 1 for examples. The data are collected through a form embedded in the weekly email. The prompts change depending on what design thinking mode the teams are working on. For example, the program calls on participants to focus on prototyping in July and August, so the prompts then focus on collecting feedback. Periodically a snapshot will include multiple prompts.

The quantitative entries from individuals are summed to form a team behavior index for each mode. The qualitative entries are collected and shared with the coach every month. In addition to collecting the snapshots, all participants were given the creative agency/creative growth mindset survey (Dweck 2000, 2006; Royalty et al. 2012) before and after the initial design thinking training.

The coaches, evaluate their teams every 3 months using both the results from the qualitative entries of the snapshots and their general experience with the team. Coaches rate output for each phase of their design thinking process (empathy,

Table 1 Snapshot prompts

Prompt	Numeric question	Short answer question
Empathy	Number of users spoken to?	What did you learn?
Prototype	Number of prototypes created or iterated on?	What are you testing?
Test	Number of people tested prototypes with?	What did you learn?
Collaboration	How in sync is your team?	How could it change?

Table 2 Expert coach evaluation measures

Category	Question 1	Question 2	Question 3
Point of view	How in sync are individuals around their POV?	How meaningful is their POV?	NA
Ideation	How novel are the ideas?	How meaningful are the ideas?	How wild is the wildest idea?
Prototype	How novel are the prototypes?	How meaningful are the prototypes?	How useful are the prototypes?

define, ideate, prototype) on a scale from 1 to 6. For example, two expert coaches performed the evaluation at the end of the empathy phase. Each coach rated each team on a scale from 1 to 6 on eight total measures spanning three general categories as seen in Table 2. The two coach's scores for each measure were averaged. The average scores for each measure were then summed across all measure to form a single creative output score for each team.

We predict that teams that empathize with more customers will have a higher creative output.

The categories were designed based on two primary factors. The first factor is the overall program goals of CLP. CLP encourages the participants to drive towards solutions that are novel in the healthcare space, yet relevant and meaningful to their customers. The second factor influencing the design of the evaluation comes from accepted definitions of creativity as the production of novel and useful ideas (Amabile 1996b).

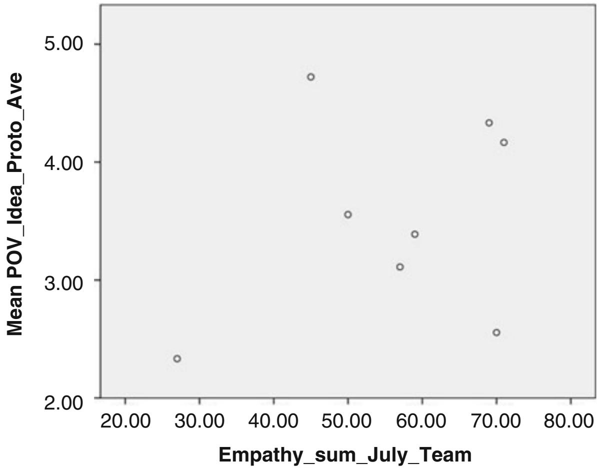
3.3 Initial Results

As this study is ongoing, we will present only the results from the first 12 weeks of the program here. This encapsulates the empathy phase where teams primarily collected human centered insights through ethnographic techniques (interviewing, observations, emersions, etc). The phase culminated with an initial point of view statement, an ideation session, and two rough prototypes. The next phase will include iterating the ideas and prototypes.

Eight teams completed snapshots for 9 weeks. There was a ninth team that we excluded from the study because they consisted of only three teammates. The first 2 weeks of the program were filled with introductions and an initial design thinking training. No snapshots were administered because the team projects had not started. The final week of the empathy phase concluded with the entire cohort meeting to share and iterate the initial concepts. Again, participants did not fill out snapshots that week.

The individual response rate was 55 % across all participants. At least one member of a team responded 89 % of the time. Most of absentees come from

Fig. 6 Empathy engagements vs. creative output



three teams, one of which had technical difficulties filling out the form from their work email addresses. All nine snapshots featured an empathy prompt. The third and seventh snapshot had a collaboration prompt as well. A prompt asking participants to evaluate their working environment was added to the fourth and eighth snapshot.

Using the grading criteria laid out in the methods section above, teams were each given a total creative output score. Figure 6 plots the total number of people interviewed in the empathy phase versus the team's creative output.

Although the number of teams (eight) makes the sample size too small to run meaningful statistics, the pattern is important to explore if we wish to test this measure on a larger sample. The second and third highest performing teams ranked third and first in empathy engagements, respectively. This is encouraging as it supports our prediction. However, two teams do not follow this pattern and it is not clear if they are outliers or not.

Another comparison between empathy engagements and creative outputs yields a stronger pattern. Figure 7 illustrates the total number of people engaged with over the last 2 weeks of the empathy phase compared with creative output.

This suggests that there might be a correlation between empathy engagements and creative output. The fact that this pattern appears stronger when constraining engagements to the final 2 weeks of the first phase may imply that continuing to practice empathy while developing a POV, generating ideas, and building prototypes has a positive effect on those tasks. Although these data are currently too limited to fully analyze statistically, the emerging patterns suggest that this measure should be explored further.

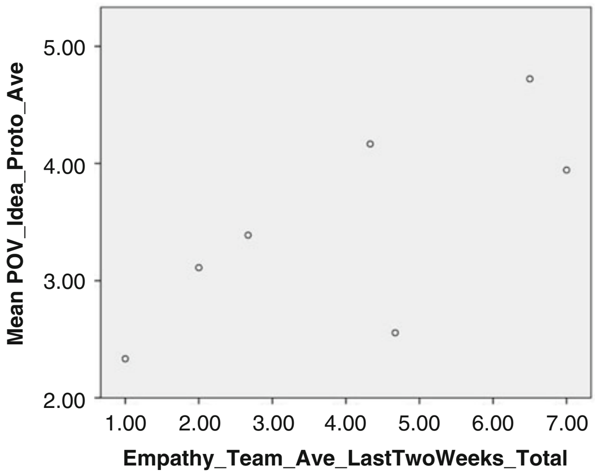


Fig. 7 Empathy engagements (last 2 weeks) vs. creative output

3.4 Limitations

The largest limitation is that this study is too small to detect any statistically significant correlations. However, as a pilot the goal is to determine if a full study should be run and what changes should be made. Another issue is the response rate. Although an 89 % team based response rate is reasonably high, the 55 % individual response rate is fairly low. The next study will either have to raise the individual response rate or place more emphasis on team response (i.e. asking each team member how many empathy engagements the entire team had). That way even if only two of four teammates fill out a snapshot, we have a stronger sense of what the team did as a whole. Finally, it is not clear if summing the three output categories to generate a total creative output is appropriate. Perhaps a more nuanced view of team performance would lead to clearer correlations.

4 Conclusion

The initial results of both pilot studies suggest that these two measures—the ecology mapping and snapshots—have the potential to accurately describe the application of design thinking in real life settings. Larger follow up studies are currently being developed.

Reflecting on feedback given by the participants in the studies, there appear to be some clear use cases for each tool. First, the ecology mapping can provide an overview of what types of projects and activities an organization uses design thinking on. Furthermore, the mapping can show how these efforts change over time. This enables leaders to more easily assess their innovation efforts and make adjustments. There is also the potential to compare design thinking strategies across companies. A core driver of the emergent communities of practice is to understand how others use design thinking. The ecology mapping allows practitioners to compare and contrast their organization with others. This could foster greater collaboration and sharing. The next step is to develop an efficient process that captures data necessary to generate ecology mappings. Then mappings for six to ten companies will be created.

The snapshots have already altered the way coaches connect with their teams. The data they provide help indicate when teams are actively engaged in design thinking. When that activity decreases coaches learn of it from the snapshot responses and can intervene. Another benefit is that all the information collected in the snapshots can be used during project reports and other storytelling settings. Teams can quickly and convincingly communicate the amount of empathy work they performed and show where the insights that drive their process came from. This is important as design thinking work may be perceived as capricious and not rigorous. As the pilot study continues it will be interesting to see how the snapshot data looks over a longer timeline.

Ultimately these measures, once developed, can be combined with individual measures of design thinking. The ability to authentically evaluate design thinking is essential to the spur further growth. Valid measures can help leaders and practitioners iterate towards stronger applications and strategies. But perhaps more importantly, a variety of metrics, mappings, and assessments can demonstrate what the impact of design thinking really is. This is the key question this movement faces.

References

Amabile TM (1996a) Creativity in context: update to “the social psychology of creativity. Westview Press, Boulder, CO

Amabile TM (1996b) Creativity and innovation in organizations. Harvard Business School, Note 9-396-239

Amabile TM, Barsade SG, Mueller JS, Staw BM (2005) Affect and creativity at work. Adm Sci Q 50(3):367–403

Csikszentmihalyi M, Larson R (1987) Validity and reliability of the experience-sampling method. J Nerv Ment Dis 175(9):526–536

Dweck CS (2000) Self-theories: their role in motivation, personality, and development. Psychology Press, New York

Dweck C (2006) Mindset: the new psychology of success. Random House, New York

Glaser BG, Strauss AL (2009) The discovery of grounded theory: strategies for qualitative research. Transaction Publishers, Piscataway, NJ

Goldman S, Carroll MP, Kabayadondo Z, Cavagnaro LB, Royalty AW, Roth B, Kim J (2012) Assessing dlearning: capturing the journey of becoming a design thinker. In: Meinel C, Leifer L, Plattner H (eds) Design thinking research. Springer, Berlin, pp 13–33

Hawthorne G, Quintin EM, Saggar M, Bott N, Keinitz E, Liu N, Reiss AL (2014) Impact and sustainability of creative capacity building: the cognitive, behavioral, and neural correlates of increasing creative capacity. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research. Springer, Heidelberg, pp 65–77

Martelaro N, Ganguly S, Steinert M, Jung M (2015) The personal trait myth: a comparative analysis of the innovation impact of design thinking tools and personal traits. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research. Springer, Heidelberg, pp 41–57

Royalty A, Roth B (in press) Developing design thinking metrics as a driver of creative innovation. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research. Springer, Heidelberg

Royalty A, Oishi L, Roth B (2012) "I use it every day": pathways to adaptive innovation after graduate study in design thinking. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research. Springer, Berlin, pp 95–105

Royalty A, Oishi L, Roth B (2014) Acting with creative confidence: developing a creative agency assessment tool. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research. Springer, Heidelberg, pp 79–96

Saggar M, Quintin E-M, Kienitz E, Bott NT, Sun Z, Hong W-C et al (2015) Pictionary-based fMRI paradigm to study the neural correlates of spontaneous improvisation and figural creativity. Sci Rep 5:10894. doi:10.1038/srep10894

Schmiedgen J, Rhinow H, Köppen E (2016) Parts without a whole?: the current state of design thinking practice in organizations, vol 97. Universitätsverlag Potsdam, Potsdam

Sonalkar N, Jung M, Mabogunje A, Leifer L (2014) A structure for design theory. In: Chakrabarti A, Blessing L (eds) An anthology of theories and models of design. Springer, London, pp 67–81

The Design Thinking Methodology at Work: Capturing and Understanding the Interplay of Methods and Techniques

Thomas Beyhl and Holger Giese

Abstract The Design Thinking methodology is one example of a design methodology that supports the creation of innovative products or services. For that purpose, the Design Thinking methodology suggests a repertoire of design phases, design activities, and design methods that can be used to solve wicked problems in terms of innovative solutions. However, since the Design Thinking methodology does not prescribe any order of design phases, activities, and methods, applications of design phases, activities, and methods lead to different shapes of the Design Thinking methodology in practice. We hypothesize that these shapes of Design Thinking at work consist of different characteristics depending on the kind of design project that has been conducted. Understanding these characteristics, their influence on the design flow itself, as well as their impact on the outcome of the design project is of major interest to managers, innovators, and researchers.

In this chapter, we report on the result of a case study that we conducted to investigate different shapes of the Design Thinking methodology in practice. As a result of our case study, we conclude that different shapes of Design Thinking methodologies exist in practice. We describe the identified characteristics and their purpose.

1 Introduction

Design Thinking is an innovation methodology that supports the solution of wicked problems in terms of innovative products or services. For that purpose, the Design Thinking methodology suggests multiple design phases, design activities, and design methods. We refer to the term design step as an umbrella word for the design phase, design activity, and design method. We refer to the term design flow as a concrete order of design steps that are used in a design project that follows the

T. Beyhl (✉) • H. Giese
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: thomas.beyhl@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_5

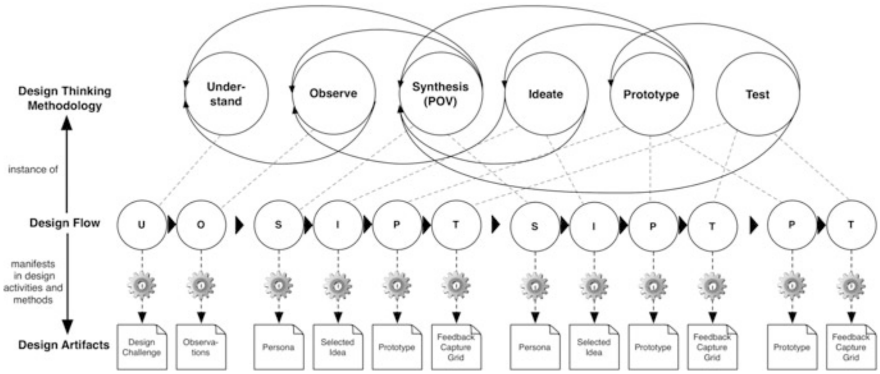


Fig. 1 Applying the Design Thinking methodology [for Design Thinking methodology cf. Plattner and Meinel (2009)]

Design Thinking methodology. Thus, a design flow represents the design journey of innovators and is an instance of the Design Thinking methodology.

However, the Design Thinking methodology does not prescribe any order to these design steps. The reason is to support innovators in choosing, with complete freedom, the most appropriate design steps for their current situation within the overall design flow. This freedom is especially needed to handle unexpected discoveries (e.g., unexpected findings) appropriately that come up during the design flow. Figure 1 depicts how the Design Thinking methodology manifests itself in practice in terms of a design flow.

A *design phase* is a step that is defined in the Design Thinking methodology. Thus, a design phase as taught at the HPI School of Design Thinking (Plattner and Meinel 2009) is an Understand, Observe, Synthesis, Ideate, Prototype, or Test step within the overall design flow. These design phases provide a meaningful segmentation of the overall design flow. This segmentation supports a common orientation among members of the design team to achieve a certain design goal and to ensure iterations of intermediate design states, using the insights gained during previous design phases. During the Understand design phase, the design team agrees on a common understanding of the design challenge. In the Observe design phase, potential stakeholders, such as prospective end users, are identified. The Synthesis design phase deals with creating a common perspective on the design problem tackled in the design project. During the Ideate design phase, ideas are created to solve the design problem. Selected ideas are prototyped in the Prototype design phase and tested during the Test design phase. Alternatively, the Understand and Observe design phase are also considered as a combined Empathize design phase (Plattner 2010). Note that these design phases are repeated iteratively in arbitrary cycles, e.g., continuing with a Prototype design phase to refine a prototype concerning the insights gained during the Test design phase or continuing with a Synthesis design phase when the Test design phase revealed that the addressed user needs are inadequate.

A *design activity* is an action that is employed to (partially) implement a design phase. Multiple consecutive design activities can be employed to implement a design phase. These design activities are either employed sequentially or in parallel when the design team splits up to investigate multiple alternatives, e.g., different prototype ideas.

A *design method* is employed to implement a design activity. Multiple design methods can be employed to implement a design activity. For example, after the testing of a prototype in the Test design phase, generally an unpacking design activity is performed using a feedback capture grid as a design method to systematically synthesize the feedback of users, who tested the prototype. This feedback concerns likes, constructive criticism, raised questions, and new ideas (cf. Plattner 2010). Note, also another design method can be employed to implement the unpacking design activity. For example, a storytelling design method can be employed to convey the experience of a user, who tested a prototype. The outcome of design methods manifests itself in *design artifacts* that are often part of project documentation. In general, these design artifacts capture the employed design flow.

Our research project was driven by the research question of whether different shapes of the Design Thinking methodology exist in practice and, if yes, what are the characteristics of these shapes.

We investigated which innovation process models and theories already exist. We summarize these models and theories as state of the art in Sect. 2. Afterwards in Sect. 3, we describe the research questions that drive our research. In Sect. 4, we present a case study that we conducted to identify different shapes of the Design Thinking methodology in practice. We summarize our research results and outline the horizon of future work in Sect. 5.

2 State of the Art

As a result of a literature survey, we identified three main research areas that are related to our investigation of the Design Thinking methodology in practice. In Sect. 2.1, we describe existing innovation process models and theories. Furthermore, we summarize our own research about capturing, recovering, and tracing the design flow of innovation projects in Sect. 2.2. Moreover, design team behavior may impact design decisions and, therefore, may influence the overall design flow. In Sect. 2.3, we describe research about design team behavior.

2.1 Innovation Process Models and Theories

In scientific literature different innovation process models and theories exist. For example, in Meinel and Leifer (2011) the Design Thinking methodology is described as a *chaotic model* that consists of five major iterative steps, namely (re)defining the problem, need finding and benchmarking, ideate, prototype, and test. The authors describe Design Thinking as "*learning through rapid conceptual prototyping*" (Meinel and Leifer 2011).

In Plattner and Meinel (2009) a *didactic model* of the Design Thinking methodology is described that consists of six design phases, namely understand, observe, define point of view, ideate, prototype, and test. A similar didactic model is described in Plattner (2010) that consists of five design phases and combines the Understand and Observe design phases into one single design phase called Empathize.

Brown (2009) describes innovation as "*system of overlapping spaces rather than a sequence of orderly steps*." He describes these spaces as inspiration, ideation, and implementation that can be passed iteratively. He says that "*Design Thinking is an exploratory process*" (Brown 2009) that can lead to unexpected discoveries along the design journey.

Lindberg et al. describe the Design Thinking methodology as "*a broad problem solving methodology that is as such no process, but shapes processes*" (Lindberg et al. 2008) motivated by the fact that didactic models (e.g. Meinel and Leifer 2011; Plattner and Meinel 2009; Plattner 2010) "*entail a certain danger of misinterpretation when they are interpreted too orthodoxly*" (Lindberg et al. 2008). Therefore, Lindberg et al. (2008) suggest an alternative conceptualization that consists of *working modes* and *working rules* to avoid traditional process terminology. They describe the Design Thinking methodology as an iterative alignment of exploring the problem space and solution space, which manifests itself in a design workflow.

Skogstad and Leifer (2011) describe innovation as unexpected discoveries and propose the Unified Innovation Process Model for Engineering Designers and Managers. The Unified Innovation Process Model describes actions of designers and managers. We learn how their actions consequently affect what the other party does. While managers steer the overall process by allocating resources and ensuring valuable outcomes for shareholders, designers generate ideas, test them, and refine them until they are ready for production. However, when managers block resources, they indirectly hinder innovation, because designers cannot build and test their ideas to make unexpected discoveries. On the other hand, it is hard for designers to justify their actions to managers when they cannot build and test their ideas. As stated by Skogstad et al. "*designers have limited ability to plan for insight discovery*" (Skogstad and Leifer 2011). The Unified Innovation Process Model describes the kernel of all phases of design flows to understand the other party's behavior. The process model consists of the iterative activities: plan, execute, and synthesize. There are also several interruption points in between that enable designers and managers to interact with each other.

Edelman and Leifer (2012) describe designing as a path determination that consists of way finding and navigation. Way finding deals with "*making significant changes to an object*" (Edelman and Leifer 2012), while navigation deals with "*making incremental changes to an object*" (Edelman and Leifer 2012). Way finding and navigation are parts of the design flow.

2.2 Capturing, Recovering, and Tracing Innovation

In our past research, we argued that innovation processes should support traceability (Beyhl et al. 2013a, c) to enable engineers to implement outcomes of innovation processes in a feasible, desirable, and viable manner. We proposed a recovery approach that searches for patterns within the documentation of design projects to recover the design journey of innovators by creating traceability links between design artifacts (Beyhl and Giese 2015a, b, 2016).

Furthermore, we investigated together with the HPI School of Design Thinking how students can be supported in documenting their design projects. As result, we implemented and evaluated one digital software tool and one analog paper tool. The digital software tool, called ProjectZoom (Beyhl et al. 2013b), enables students to document their design projects using a virtual whiteboard. ProjectZoom aggregates captured design artifacts from multiple digital sources. Students are thus able to cluster and interrelate these artifacts by drawing circles around artifacts, drawing lines between artifacts and created clusters, as well as adding textual annotations to artifacts, lines, and circles. This kind of documentation can show the employed design flow.

The analog paper tool is called LogCal (Menning et al. 2014) and enables the template-based documentation of design projects. The LogCal employs Plan-Do-Check-Act (PDCA) cycles to support students in creating and retrieving design rationales, reflecting on their design flow, and compiling final documentation of their design project. The LogCal consists of multiple pages. Each page employs a PDCA cycle by providing text boxes that ask students to document what they plan for the current working day (Plan), how they realize their plans (Do), what results they achieve (Check), and how they plan to proceed (Act). Furthermore, the LogCal asks students to reflect on their design flow. The LogCal provides a means of capturing the employed design flow.

2.3 Design Team Behavior

In general, two kinds of design team behavior analysis exist. Either the design teams are observed in real-time, i.e., online, with feedback provided to the design team immediately, or the design team behavior is analyzed retrospectively, i.e., offline, without the design team receiving behavioral feedback. The latter enables more in-depth research in order to reveal certain patterns of design behavior.

Kress and Sadler (2014) propose TeamSense to support team dynamic measurements. TeamSense is a “distributed network of unobtrusive, ambient sensors to measure team function in real time” (Kress and Sadler 2014). With the help of TeamSense the authors aim to accelerate the collaborative design flow by employing unobtrusive sensors in design workspaces. These sensors detect patterns of team activity in an event stream of measurements, and provide dynamic feedback to design teams based on the real-time measurements.

Sadler and Leifer (2015) are developing TeamSense further in terms of a prototyping toolkit that enables the design team, design coaches, and managers to get more insights about design team performances. These insights may be used to improve design team performance.

Sonalkar et al. (2016) propose the Interaction Dynamics Notation (IDN), which is a diagnostic instrument that enables to analyze design team interactions and behaviors that influence design outcomes. With IDN the authors aim to isolate interaction behaviors of design teams, improve design team performance, and, in the long run, better the design outcomes that result.

The IDN instrument motivated the development of additional notations for the analysis of design team behavior. Scheer et al. (2014) propose the Knowledge Handling Notation (KHN) that aims at enabling design conversation diagnosis and identifying patterns of knowledge handling. The authors state that knowledge handling, e.g., in terms of design reviews, is one way to bridge design phases. Menning et al. (2015) propose the Topic Markup Scheme (TMS). It enables capturing and representing "*the topical structure of a conversation in the form of topic threads*" (Menning et al. 2015). IDN, KHN, and TMS provide three lenses that promise new insights on design team behavior concerning interaction, knowledge handling, topic treatment, and topic alignment within design teams.

3 Research Question

The state of the art in Design Thinking Research and innovation processes employs different *microscopic* lenses to investigate the interplay of design team decisions, behaviors and interactions. These microscopic lenses focus on certain aspects within design projects. In addition, our research aims at investigating the whole design flow from the initial design challenge to the final innovative outcome of the design project. We consider our research on employed Design Thinking methodologies as a *macroscopic* lens that focuses on the design flow as a whole. Our research is driven by a desire to consider microscopic lenses and macroscopic lenses as complements. The combination of microscopic and macroscopic lenses supports a better understanding of the Design Thinking methodology in practice.

Existing Design Thinking research focuses on microscopic lenses. At the time of writing no research is known to us that investigates the concrete order of design phases, design activities, and design methods in practical Design Thinking projects. Only idealized Design Thinking process blueprints exist. These blueprints rarely convey applied design flows from real Design Thinking projects.

As a result of our literature survey, we hypothesize that different shapes of the Design Thinking methodology are at work in terms of applied design flows. We further hypothesize that these shapes of design flows consist of different characteristics, which depend on the aim and scope of the design project. For example, from observations during our former research projects about traceability for innovation processes (Beyhl et al. 2013a; Beyhl and Giese 2015a, 2016), as well as existing innovation process models, (Lindberg et al. 2008) we hypothesize that experiment-centered and divergence-centered shapes of the Design Thinking methodology exist. The question that arises is whether more shapes of the Design Thinking methodology exist and, if so, what are their characteristics.

In our research project, we aim at identify these different shapes of design flows and their characteristics by extracting them from conducted design projects to base our conclusions on real data about design projects in practice.

4 Case Study

We conducted a case study to answer our research question of whether different shapes of Design Thinking methodologies exist in practice and which characteristics these shapes have. As described in Sect. 1, applying the Design Thinking methodology manifests itself in design artifacts that are explicit outcomes of design methods and, therefore, capture the overall design flow. For our case study, we analyzed these design artifacts to recover the design flow as depicted in Fig. 2. The recovery of the design flow itself is a challenging task, because the same design methods can be employed during different design activities and same design activities can be employed during different design phases. Therefore, the recovered design flow may differ from the design flow that is actually employed.

In Sect. 4.1, we describe the design of our case study. We present the recovered design flows in Sect. 4.2. In Sect. 4.3, we interpret the recovered design flows. Finally, we comment on the validity of our case study in Sect. 4.4.

4.1 Case Study Design

In our qualitative case study we investigated ten educational Design Thinking projects conducted at the HPI School of Design Thinking[1]. Each project had a duration of 6 weeks (12 working days) and consists of a project documentation. The project documentation consists of design artifacts such as photographs of post-it walls and outcomes of employed design methods. These design artifacts are organized in terms of file shares. Each design team was solely responsible for determining which design outcomes are documented and how they are documented.

We selected 10 out of 48 design projects for manual design flow recovery from 2013 to 2015. We selected design project documentation that consisted of at least one-level file share hierarchy. Furthermore, the design project documentation had

[1]http://hpi.de/school-of-design-thinking.html (last access: October 15th 2015).

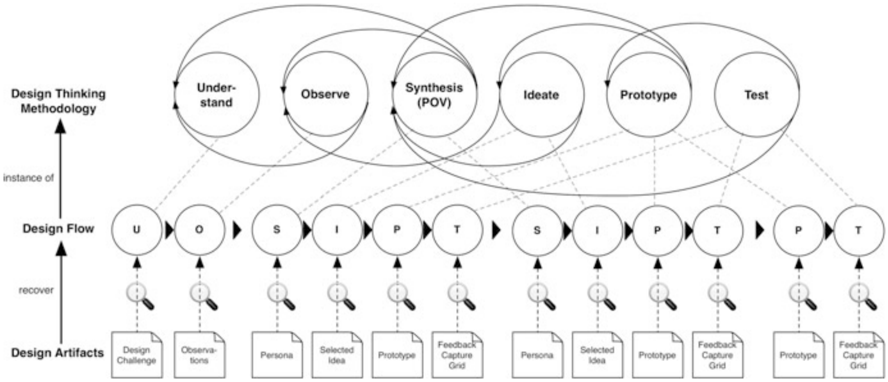


Fig. 2 Recover the employed design flow from design artifacts

to be visibly complete and evenly distributed over the years. Moreover, we included three projects that were investigated for a former traceability recovery experiment (cf. Beyhl and Giese 2015b).

The recovery of the design flow based on design documentation was performed objectively by one single researcher in the research team. The design flow was recovered manually (i.e., without any software tool support) by investigating the documented design artifacts. Additionally, analog documentation tools provided by HPI School of Design Thinking to support students in documenting their projects were investigated. For example, the LogCal (Menning et al. 2014), which enables the template-based documentation of design projects, was investigated to confirm the recovered design flow. Furthermore, quick response codes (QR codes) were exploited in two projects. In these cases students attached QR codes to post-it walls and outcomes of design methods before taking photographs for documentation purposes to support traceability for their design projects (Beyhl et al. 2013a).

4.2 *Recovered Design Processes*

In this section, we present the raw design flows that we recovered from the ten selected design projects. Figure 3 shows the recovered design flows. The top of Fig. 3 shows the legend of our notation. Solid circles denote the design phases. Dotted circles depict skipped or undocumented design phases. The letters in the circles denote the design phases Understand, Observe, Empathize (the combined Understand and Observe design phases), Synthesis, Ideate, Prototype, and Test. The design phases are depicted in chronological order bottom up. Dashed rounded rectangles mark iterations of design phases. Parallel tracks denote divergent and convergent design phases.

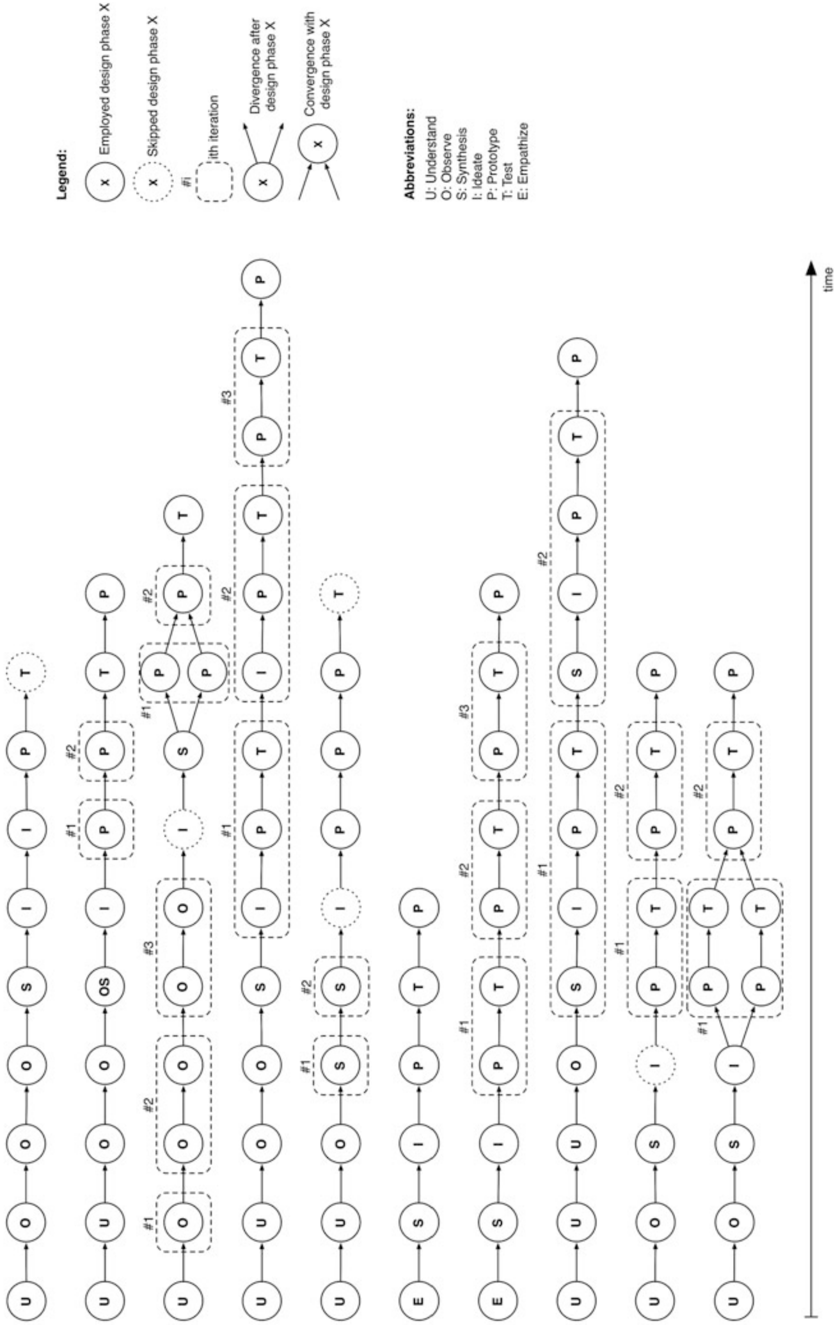


Fig. 3 Recovered design phases, including iterations and divergent/convergent design phases

The projects one to three were conducted during winter term 2013. The projects four to six were conducted during summer term 2014. The projects seven to ten were conducted during summer term 2015.

All recovered design flows start in a similar way by employing an Understand and Observe design phase. The projects six and seven combine the Understand and Observe design phase in a single Emphasize design phase. The processes mainly start to differ after the Observe phase. They differ in the number of iterations, the design phases that are part of these iterations, and whether divergent and convergent design phases exist. Furthermore, each design project either ends with a Prototype or Test design phase.

For example, project four started with two subsequent Understand design phases, followed by two Observe design phases. Afterwards, one Synthesis design phase was employed. Then two iterations of Ideate, Prototype, and Test design phases were employed. Subsequently, one iteration was conducted that consisted of one Prototype and one Test design phase. The design flow ended with one single Prototype design phase.

Table 1 shows employed design activities, design methods, as well as created design artifacts in each design phase. We used the design artifacts to argue about employed design methods and design activities. We used the recovered design activities to conclude which design phase was conducted. The numbers in brackets denote the number of recovered design activities, design methods and design artifacts for *all* investigated design projects. The design activities, design methods, and design artifacts are ordered by frequency of occurrence.

For example, when a design project was kicked off the design challenge was discussed, the prospective design flow was planned, and a fast-forward design activity was employed. During these design activities the design methods: mind map, analogies, and stakeholder map, were employed. The design methods resulted in mind maps, initial point-of-view statements, and initial ideas that addressed the design challenge.

4.3 Case Study Interpretation

In this section, we interpret the recovered design flows and investigate commonalities and differences. We clustered the recovered design flows concerning their general structure, such as order of design phases and kinds of design phase iterations, to interpret the recovered design flows in a uniform way. Figure 4 depicts the recovered shapes of the Design Thinking methodology. The recovered design flows prove that design flows differ between different design projects. We identified the following shapes of the Design Thinking methodology.

First, we identified design flows that do not consist of iterations or design steps that are outstanding. Therefore, these design flows have a waterfall-like shape (cf. DF1).

Table 1 Recovered design activities, design methods, and design artifacts

Design step	Understand	Observe	Synthesis	Ideate	Prototype	Test
Design activities	Kick-off (6) Challenge discussion (6) Process planning (5) Fast forward (5) Interview project partner (4) Team building (3)	Visit stakeholder (4) Interview preparation (3) Planning (3) Prototype (1) Unpacking (1)	POV (8) Unpacking (3) HMW (1) User needs (1)	HMW (4) Refining POV (2) Brainstorm (2) Idea selection for prototyping (2) Idea pitch (1)	Sketching prototype (4) Prototyping (2) HMW recap (2)	Unpacking (4) Planning (1) POV recap (1) Team building (1)
Design methods	Mindmap (4)	Clustering (4)	Clustering (3)	Clustering (5)	Paper prototype (3)	Feedback grid (6)
	Analogies (4)	Stakeholder observation (3)	Venn-diagram (2)	Voting (5)	Lego prototype (3)	Mindmap (1)
	Stakeholder Map (3)	Stakeholder map (3)	User journey (1)	Brainstorm (4)	Role play (1)	
	Brainstorm (3)	Contradiction/surprise (3)	Storyboard (1)	H2-Questions (2)	Problem safari (1)	
	POV (3)	Storytelling (2)	Voting (1)		Customer/user journey (1)	
	Time table (2)	Review scientific articles (2)	Mindmap (1)		Idea pitch (1)	
	Clustering (2)	Interview checklist (1)	HMW (1)			
	H2-Questions (2)	Mindmap (1)	Two mindsets (1)			
	Voting (1)	Wizard of Oz prototype (1)				
	Challenge dimensions (1)	H2-Questions (1)				
	HMW (1)	Voting (1)				

(continued)

Table 1 (continued)

Design step	Understand	Observe	Synthesis	Ideate	Prototype	Test
Design artifacts	Mindmap (4) POV (3) Idea parking space (1) Interview protocol (1) Assumptions (1)	Interview protocol (3) Empathy grid (3) User needs (2) Mindmap (1) Tag cloud (1) Photos of stakeholders (1)	Persona (5) Stakeholder map (3) Mindmap (3) Feedback grid (2) HMW-Questions (1) User needs (1) Assumptions (1)	POV (2) Pilot prototype (2) Idea dashboard/sketch (4)	Physical prototype (6) Prototype storyboard (2) Idea dashboard (1) Mindmap (1)	Feedback grid (6) Quotes (1)

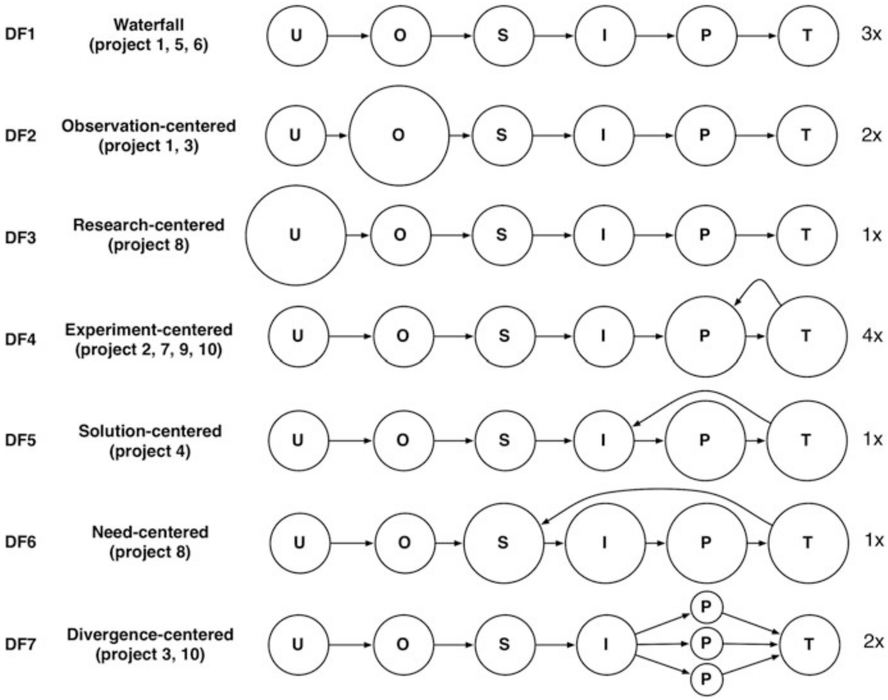


Fig. 4 Shapes of the Design Thinking methodology in practice

Furthermore, we identified design flows that consist of remarkable Observe design phases, i. e., either multiple iterations with Observe design phases or one long Observe design phase. We characterize these design flows as observation-centered (cf. DF2). Examples of this observation-centered design flow can be seen in design project one and three.

In contrast to observation-centered design flows, we identified research-centered design flows (cf. DF3), which consist of remarkable Understand design phases. For example, we consider design project eight as research-centered.

The majority of the recovered design flows has the shape of an experiment-centered design flow (cf. DF4). Experiment-centered design flows are characterized by multiple iterations that consist of Prototype and Test design phases. These kinds of iterations show that the user needs identified during the Synthesis design phase are generally appropriate and that the design solution the design team is aiming for is promising and addresses the design challenge. For example, the design projects two, seven, nine, and ten consist of an experiment-centered shape of the Design Thinking methodology.

Deviations of the experiment-centered design flows are need-centered design flows and solution-centered design flows. Solution-centered design flows (cf. DF5) of the Design Thinking methodology include iterations with the design phases Ideate, Prototype, and Test. Such iterations convey, that the identified user needs

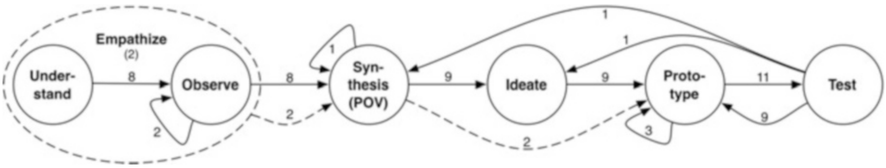


Fig. 5 Recovered Design Thinking methodology

are appropriate, because no additional Synthesis design phase is employed. However, the design team aimed at a solution that was revealed as not promising. Therefore, the design team elaborates new solution ideas based on the insights gained during the Prototype and Test design phases. For example, project four is a solution-centered design flow.

In contrast, need-centered design flows (cf. DF6) of the Design Thinking methodology include iterations with the design phases Synthesis, Ideate, Prototype, and Test. Such iterations reveal that the design team aimed at user needs that are either not the right user needs or an incomplete identification of user needs. Therefore, previous research results and observations are revised with the insights gained during the Prototype and Test design phases to refine the user needs that need to be addressed. For example, design project eight has a need-centered design flow.

Design projects three and ten consist of parallel design paths during the Prototype and Test design phases. Creating prototypes and testing prototypes in parallel are well-known as divergent and convergent design activities (cf. Lindberg et al. 2008). Therefore, we conclude that divergent-centered and convergent-centered design flows exist.

We characterize this shape of the Design Thinking methodology as divergent-centered (cf. DF7). For example, design projects three and ten are divergent-centered.

Figure 5 depicts a derived Design Thinking process model that combines all investigated design projects. The numbers attached to the solid lines denote the number of transitions from the source design phase to the target design phase for all ten projects in total. For example, the transition from the Prototype design phase to the Test design phase was passed 11 times.

The derived Design Thinking process model enables four general observations. First, two times the Understand and Observe design phases have been combined to a common Empathize design phase. Second, design phase iterations exist that consist of a single kind of design phase. For example, three iterations of the Prototype design phase exist without Test design phases in between. Third, iterations do not consist of Understand and Observe design phases. Thus, only the design phases Synthesis, Ideate, Prototype, and Test are part of iterations. Fourth, two times the Ideate design phase was skipped (or was not explicitly documented). Note, these observations are only true for the ten selected projects that are part of our case study.

4.4 Case Study Validity

In this section, we comment on the validity of our case study. We manually analyzed the design projects on our own. At least one independent person should perform the analysis as well to ensure the validity of the results.

At this time we decided to perform the analysis manually and without any automated approach (e.g., Beyhl and Giese 2015b) to eliminate reasons why the case study results may be invalid. For example, we wanted to reduce the impact of imprecise recovery algorithms. Furthermore, we wanted to create a basic truth that can be used when evaluating an automated approach.

We selected 10 out of 48 projects for analysis. It is possible that ten other projects could have led to different analysis results. Furthermore, we analyzed ten educational projects with a duration of 12 working days. Projects from business and projects with a longer duration may end up with different results.

Furthermore, the identified shapes of the Design Thinking methodology are fuzzy. Therefore, classifying the recovered design flows is a difficult task and may end up in different classifications when different people create this classification.

Finally, when the design teams are asked to reflect on their design flow, they may observe a different design flow, because they were part of the design flow. That means the internally perceived and externally observed design flow may differ, because the design teams know the rationales for their decisions within their design flow. These rationales are not always obvious to outsiders.

5 Conclusion and Future Work

In this chapter, we have shown that in practice design flows are shaped different. We revealed different shapes of the Design Thinking methodology at work and argued about their purposes. Our current state of research embodies potentialities for future work. For example, the rationales for transitions between design activities and design phases are currently not considered by our case study due to the fact that recovering these rationales is a challenging task; because these rationales often remain undocumented. Analyzing audio or video recordings of design teams at work may help to reveal such rationales. Furthermore, employing existing instruments for design team diagnosis such as IDN (Sonalkar et al. 2016), KHN (Scheer et al. 2014), or TMS (Menning et al. 2015) may help to argue about employed design flows. We plan to employ a semi-automated recovery and analysis approach for design flows based on our former work about traceability for innovation processes (Beyhl and Giese 2015b). It may be helpful to incorporate instruments for design team diagnosis to create an overall understanding of employed design flows.

Acknowledgements We thank HPI School of Design Thinking for supporting our research by providing design project documentation.

References

Beyhl T, Berg G, Giese H (2013a) Connecting designing and engineering activities. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—building innovation eco-systems, understanding innovation. Springer, Heidelberg, pp 153–182. doi:10.1007/978-3-319-01303-9_11

Beyhl T, Berg G, Giese H (2013b) Towards documentation support for educational design thinking projects. In: Lawlor J, Reilly G, Simpson R, Ring M, Kovacevic A, McGrath M, Ion W, Tormey D, Bohemia E, McMahon C, Parkinson B (eds) International conference on engineering and product design education, Dublin, Ireland. Design Society, Glasgow, pp 408–413

Beyhl T, Berg G, Giese H (2013c) Why innovation processes need to support traceability. In: Proceedings of 7th international workshop on traceability in emerging forms of software engineering. IEEE, San Francisco, pp. 1–4. doi:10.1109/TEFSE.2013.6620146

Beyhl T, Giese H (2015a) Connecting designing and engineering activities II. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—building innovators, understanding innovation. Springer, Heidelberg, pp. 211–239. doi:10.1007/978-3-319-06823-7_12

Beyhl T, Giese H (2015b) Traceability recovery for innovation processes. In: Proceedings of the 8th international symposium on software and systems traceability. IEEE, Piscataway, pp. 22–28. doi:10.1109/SST.2015.11

Beyhl T, Giese H (2016) Connecting designing and engineering activities III. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—making design thinking foundational, understanding innovation. Springer, Heidelberg, pp 265–290. doi:10.1007/978-3-319-19641-1_16

Brown T (2009) Change by design. HarperCollins, New York

Edelman JA, Leifer L (2012) Qualitative methods and metrics for assessing wayfinding and navigation in engineering design. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—measuring performance in context, understanding innovation. Springer, Heidelberg, pp 151–181. doi:10.1007/978-3-642-31991-4_9

Kress G, Sadler J (2014) Team cognition and reframing behavior: the impact of team cognition on problem reframing, team dynamics and design performance. In: Leifer L, Plattner H, Meinel C (eds) Design thinking research—building innovation eco-systems, understanding innovation. Springer, Heidelberg, pp 35–48. doi:10.1007/978-3-319-01303-9_3

Lindberg T, Jobst B, Gumienny R (2008) Is there a need for a design thinking process? In: Design thinking research symposium 8. DAB Documents, Sydney, pp 243–254

Meinel C, Leifer L (2011) Design thinking research. In: Plattner H, Meinel C, Leifer L (eds) Design thinking understand—improve—apply, understanding. Springer, Heidelberg, pp xiii–xxi

Menning A, Beyhl T, Giese H, Weinberg U, Nicolai C (2014) Introducing the LogCal: template-based documentation support for educational design thinking projects. In: Proceedings of the 16th international conference on engineering and product design. Design Society, Glasgow

Menning A, Scheer A, Meier BH, Nicolai C (2015) Designing as weaving topics: coding topic threads in design conversations. In: Proceedings of the international association of societies of design research. IASDR, Brisbane

Plattner H (2010) Bootcamp bootleg. Institute of Design at Stanford. http://dschool.stanford.edu/wp-content/uploads/2011/03/BootcampBootleg2010v2SLIM.pdf. Accessed 14 Oct 2015

Plattner H, Meinel C (2009) Design thinking—Innovation lernen—Ideenwelten öffnen. mi-Wirtschaftsbuch, Finanzbuch Verlag GmbH, München

Sadler J, Leifer L (2015) TeamSense: prototyping modular electronics sensor systems for team biometrics. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—building innovators, understanding innovation. Springer, Heidelberg, pp 87–100. doi:10.1007/978-3-319-06823-7_7

Scheer A, Menning A, Helldorff E, Rhinow H, Nicolai C (2014) The knowledge handling notation: building an interface to enable design conversation diagnosis. In: Proceedings of the design thinking research symposium 2014. Purdue University, West Lafayette. doi:10.5703/1288284315945

Skogstad P, Leifer L (2011) A unified innovation process model for engineering designers and managers. In: Meinel C, Leifer L, Plattner H (eds) Design thinking—understand—improve apply understanding Innovation. Springer, Heidelberg, pp 19–43. doi:10.1007/978-3-642-13757–0_2

Sonalkar N, Mabogunje A, Pai G, Krishnan A, Roth B (2016) Diagnostics for design thinking teams. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research—making design thinking foundational, understanding innovation. Springer, Heidelberg, pp 35–51. doi:10.1007/978-3-319-19641-1_4

On Creating Workspaces for a Team of Teams: Learnings from a Case Study

Marie Klooker, Claudia Nicolai, Stephan Matzdorf, Arne Trost, Karen von Schmieden, Lilith Böttcher, and Ulrich Weinberg

Abstract Workspaces provide resources, facilitate (different) modes of working, and communicate an organization's culture. As such they are a powerful resource to amplify the innovation culture of an organization.

Changemakers and leaders of organizations face the challenge to create workspaces that promote innovative activities of teams. Companies too often copy well-known best practice examples and the physical design of the workspace is put into focus. Organizational structures and context are disregarded and the inherent needs of an organization and employees are neglected. As a result, the workspace is used differently as initially intended and expectations are not met. A methodological approach to the creation of workspaces that fit the needs of their users and managers is still needed.

Impact Hub Berlin—a co-working space for social entrepreneurs, serves as example of a workspace that amplifies the collaborative culture of its community. The approach to the creation of its new workspace involves the integration of users from the beginning and, thus, the creation of a "shared ownership." This concept, which can be applied to a wide range of contexts, provides the basis for a well-functioning, innovative work environment in line with specific organizational needs and objectives. This article provides an overview of insights gained through the exploration of the case.

1 Introduction

Globalization and the increase of interconnectedness through innovations in the fields of technology, communication and transportation change our world drastically and raise great challenges for people and organizations. This results in trends

---

The original version of this chapter was revised. An erratum to this chapter can be found at DOI 10.1007/978-3-319-40382-3_18.

M. Klooker (✉) • C. Nicolai • S. Matzdorf • A. Trost • K. von Schmieden • L. Böttcher • U. Weinberg
HPI School of Design Thinking, Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: marie.klooker@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_6

of dramatically increasing complexity (IBM 2010, p. 15) and speed (Mootee 2013, p. 3)—in particular at the workplace of knowledge workers. Instead of having one clear task, knowledge workers are confronted with a great variety of them to be managed simultaneously. Daily work is now characterized by fragmentation and disruption. With the opportunities provided by the information technology the role of the workspace is changing, and a fixed physical place is no longer necessary. At the same time, the well-educated and highly demanding 'Generation Y' is entering the labor market, for whom "the prospect of an entire working life within the same organization is more a nightmare than a life's dream [and who] want to develop, learn, work project-based and in parallel." (Dark Horse Innovation 2014, p. 23)

Organizations have to adapt accordingly and "establish effective workspaces in order to enable appropriate spatial interactions as well as to create physical environments for diverse innovation activities" (Moultrie et al. 2007, p. 53). Peters (1992, p. 413) goes even further and depicts space management as the most ignored and simultaneously the most powerful tool for implementation of cultural change and to foster innovation and learning within organizations.

However, well-grounded academic research on the attributes of effective work environments to support creativity and innovation is still missing. It "appears that firms create spaces based on instinct and personal judgment, rather than on genuine insights based on firm evidence" (Moultrie et al. 2007, p. 54). Persons in charge of establishing innovation spaces tend to neglect an analysis of the company's and employees' needs and interaction patterns; concepts are often copied from other companies meanwhile ignoring the own innovation culture. Each organization and its people have individual requirements. This means it is not sufficient to create an innovative ambiance by copying 'best' aesthetic practices from other 'great places to work' (Friedman 2014; Moultrie et al. 2007). This raises the question of how organizational leaders and change-makers can account for this. We try to tackle the problem by investigating how innovation spaces are created within organizations in order to identify patterns matching spatial structures with work processes in the field of innovation activities. By means of case-study analysis, we examine the network organization Impact Hub Berlin, as a best practice example. We seek to answer the following research question: What defines the spatial structures (physical/socio-organizational) of Impact Hub Berlin with regard to its initial strategic intent for creating the workspace? In this context, we apply Moultrie's et al. (2007) model of creating innovation spaces as expressed in the practices of processual guidance and theoretical orientation.

2 Innovation Spaces in Organizations: Providing the Framework

Spatial factors that support the creative actions of teams have been identified in previous research (e.g. Amabile and Gryskiewicz 1989; Deb and Sinha 2011; Dul et al. 2011). However, when it comes to an organization's capacity to foster innovation, the workspace needs to be viewed in terms of the work environment as a whole (Weinberg et al. 2014). The workspace entails physical as well as socio-organizational structures that need to be understood within their specific context.

Amabile's componential model for organizational innovation (1988; Amabile et al. 1996) provides the theoretical framework for evaluating the work environment in regard to its potential for innovation capacity building.

The model displays the link between the work environment, individual creativity, and organizational innovation (cf. West and Sacramento 2012). It shows how the work environment affects the relevant components that influence individual creativity. Amabile identified three crucial aspects when it comes to creativity: domain relevant skills in terms of factual knowledge and expertise; creativity-relevant skills, including strategies and cognitive styles that influence the idea generation, and a person's intrinsic motivation expressed as the genuine interest in a task (Amabile 1988). The latter is regarded as key when it comes to enhancing creative capacity and innovation since, as Amabile (1983) points out, intrinsic motivation differentiates between what one can do and what one actually will do.

Accordingly, an organization's capacity for innovation is defined by its motivation to innovate, prevalent management practices that facilitate innovation within the organization and any resources provided to support creativity and innovation at work (Amabile 1988; Amabile et al. 1996). As a compound all three aspects shape the work environment of an organization. Consequently, we conclude that organizational innovation capacity is fostered by workspaces involving the three functional dimensions aligned towards innovation. First, an innovation space facilitates skills and work modes that are put into practice during the innovation process. Second, it provides physical and nonphysical resources in the form of knowledge and/or tools and materials that support innovation. Third, it communicates an image of the organizational culture that expresses its motivation to innovate (see Fig. 1).

2.1 *Organizational Space*

The concept of space is rather complex and elusive. In line with the spatial turn in organizational and cultural science (cf. Guenzel 2008) we apply a relational understanding of 'space' in our examination of the organizational workspaces.

Throughout the past centuries the concept of space developed from an absolute to a relative. Both were opposing for a long time. At the beginning of the twentieth century, especially with the theory of relativity, scientists realized that neither time

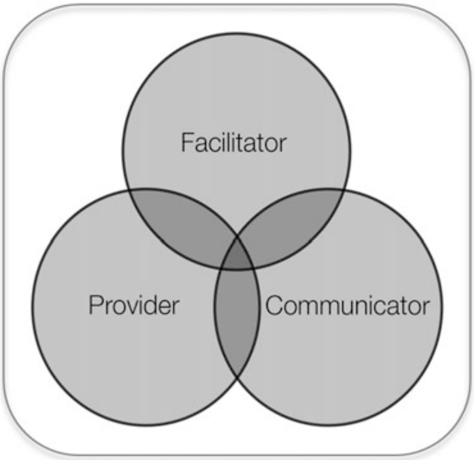


Fig. 1 Three conceptual dimensions of innovative workspaces in organizations

nor space is static leading to perception of space as being unsteady and in constant flux, until, in line with the so-called *spatial turn* (in the late 1980s), new concepts of space where developed, adding to the debate in the social sciences. (cf. Guenzel, 2008)

For our conceptual approach we use Martina Löw's (2007) relational theory of space. As relational concept, space is socially constructed and formed by acts, interactions and constellations of human beings. Löw's (2007) theory forms the basis to investigate the effects of space on teams and innovation processes. Introducing time as flexible dimension, space is no longer a rigid container. Moreover it is interconnected to the world of physical goods (Löw 2007; von Weizsäcker 1990). "The space depends on the reference system of the observer" (Löw 2007, p. 34), and thus becomes a context dependent and perpetually evolving and adapting concept. Löw further defines space as being a twofold phenomena, consisting of a "relational array/order of living organisms and social goods. Two processes to be differentiated analytically constitute space: spacing and synthesis" (Löw 2007, p. 159). Spacing describes the construction and positioning of objects, whereas perception, imagination and memory synthesize the space. The synthesis sums up human beings and social goods to spaces. Thus "space derives from the interaction of structure and action" (Löw 2007, p. 53) and is a co-creation of this reciprocity during its process of creating as illustrated in Fig. 2.

Recent theory of space is based on the findings of Henri Lefebvre, described in "La production de l'espace" (1974). Lefebvre describes (urban) space as differential space that is created by various social and functional elements. Lefebvre differentiates between perceived space, conceived space, and lived space (Lefebvre 1974). In comparison to Löw's (2007) aforementioned traits of space, Lefebvre neglects the impact of time on space and rather analyses different attributes of space existing simultaneously and overlapping one another. We assume that the close interlocking of the strategic level (conceived space) and the tactical level (perceived space) can promote the production of effective innovation spaces.

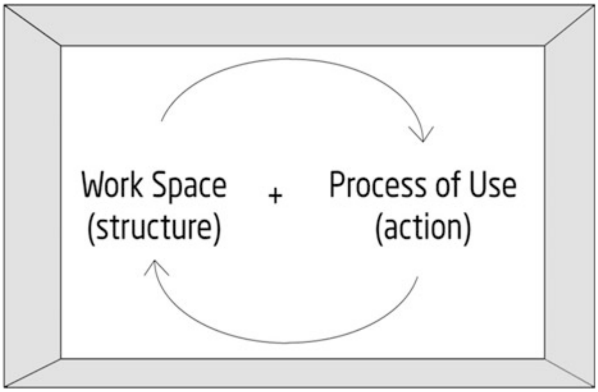


Fig. 2 Space as a relational concept

Linking Lefebvre's understanding of the production of space to architectural space intended to promote innovation, Kornberger and Clegg (2004) assume: Effective innovation spaces mirror the characteristics of 'generative buildings'. Opposing terminal architecture, in which form follows function and buildings are ordered, inflexible, uncommunicative and static, this conceptualization offers a balance of predictability and randomness, flexibility, communication resulting in problem generation and movement. Generative buildings organize the flows of communication, knowledge, and movement, which may result in the emergence of 'surprises'. Furthermore, such facilities offer positive power of spatial organization and are places where inhabitants are 'illegal architects' who (ab)use and (re) define space. In line with Kornberger and Clegg (2004) we state that organizing the generative building means producing space that informs the space of innovation.

3 Research Design

We have chosen an exploratory case study approach to explore the area of collaborative workspace design. The case we examine is a co-working space for social entrepreneurs, the Impact Hub Berlin. We selected this case because it enabled us to examine the development process of an innovative co-working space with a collaborative approach at its diffferent stages over time (longitudinal research).

The case study is designed based on the transitional framework developed by Moultrie et al. (2007). It covers the process of innovative workspace development in organizations. The model provides the basis to compare and evaluate workspaces created within organizations. In its original version, it focused on physical space. Based on our working definition and our research findings, we iterated Moultrie's model and replaced 'physical space' with 'work space' (Fig. 3):

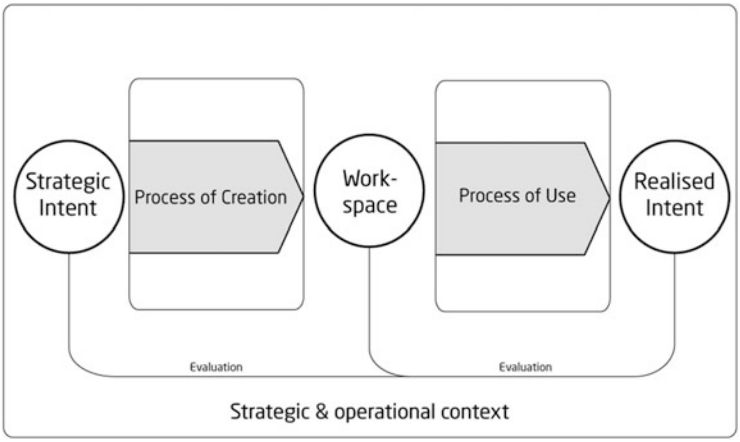


Fig. 3 Process of workspace development based on Moultrie et al. (2007)

3.1 Field-Work: Document Analysis and Qualitative Interviews

Our research is qualitative in nature, relying on the analysis of documents and interviews. In a first step, aimed at exploring the strategic intent preceding the workspace design, we conducted three semi-structured interviews with all three members of the management team, combining the flexibility of open-ended interviews with the intended directionality of the survey instrument. In constructing questions for the interview guideline, we followed the recommendations of Schensul et al. (1999, p. 154). The three interviewees are the co-founders of the Impact Hub Berlin and responsible for management, program and the process for spatial redesign. Consequently, we analyzed documents provided by the Impact Hub founding team, covering the collaborative design process. To gain a deeper understanding of the process of creating the new workspace, we also conducted a roughly 45 min group interview with all three founders together. Each interview lasted between 35 and 45 min and was recorded, transcribed, and analyzed by means of qualitative content analysis (cf. Mayring 2008). In a third step, we analyzed the designed structures of the workspace by means of on-site visits and non-participatory observation.

3.2 Operationalization of Key Concepts

Before clarifying the overall approach taken in this paper, the relevant key concepts have to be defined: *Creativity* is understood as generating ideas that are new and useful to the organization (Amabile 1988; George and Zhou 2002; Tierney and

Farmer 2002). "(*Innovation*) implicitly or explicitly includes the motion of creative ideas being successfully implemented by a larger group" (Amabile 1988, p. 127). Innovation refers to the implementation of such ideas (Amabile 1988; Anderson et al. 2004). Accordingly we apply the definition of *work-environment creativity* as "the tendency of employees within an individual work environment to produce novel ideas that are useful in an organization" (Schepers and Van den Berg 2007, p. 2; cf. Amabile et al. 1996; Stein 1974; Woodman et al. 1993).

The concept of *strategic intent* originates from management studies. Hamel and Prahalad (1989, 2005) define it as "a [sustained] obsession with winning at all levels of the organization" (Hamel and Prahalad 2005, p. 150). Strategic intent goes beyond the strategic planning of deliberate actions and changes and entails objectives 'for which one cannot plan' (emergent phenomena) (ibid.). Building upon this, different scholars have transferred the concept of strategic intent to the academic field of organizational strategy. It is directed towards the future and resembles an organization's intention by proactively activating all organizational levels for a common purpose (ibid.). Hence strategic intent serves the different parties within an organization as common, moving target to be reached in the future.

Organizational workspaces entail physical and socio-organizational structures. Structures are understood as any component of the organization that can be changed consciously or are designed with the intention to achieve a certain outcome, such as the facilitation of a specific action. Spatial factors can be categorized into the three-dimensional model, as introduced above: space as provider of resources, space as facilitator of processes as well as space as communicator of a (work) culture. For our conceptual approach, we apply a relational notion of space. The categories are to some extent overlapping because they are subject to ongoing sense-making processes by organizational members (Weick 1995). A definite categorization can only be done when taking into consideration not only the spatial factors, as designed structures within the space, but also the way they are actually interacted with by the users.

Moultrie's et al. (2007) stance on creating space serves as our conceptual approach and guidance. In this context, we structure and present the data and findings in accordance with his suggested stages of: first—strategic intent, second—process of creation and third—the space created. Statements are invoked matching the three levels of process supporting the overall assumption of Impact Hub being a space to foster and enable innovation by the means of collaboration.

4 A Case-Study Example: Impact Hub Berlin

4.1 Impact Hub Global

Impact Hub Berlin is part of the Impact Hub Association, a global collaborative network with currently 66 local Impact Hubs and more than 10,000 active members worldwide. The overall vision of the global Impact Hub community is to create positive impact around the world. Yet, "impact cannot happen in isolation. It requires collective action [. . .] through the combined accomplishments of creative, committed and compassionate individuals focused on a common purpose" (Impact Hub Global).

Collective action and the subsequent impact are cultivated in each Impact Hub according to three core elements: first, an connecting people from diverse backgrounds to create a community of social and sustainable entrepreneurship. Second, stimulating events provide the opportunity to learn and improve skills needed to achieve positive impact. Third, providing physical space, facilitating collaboration through a "flexible and highly functional infrastructure" (Impact Hub Global).

4.2 Impact Hub Berlin

As part of the global Impact Hub Association, Impact Hub Berlin launched its new co-creation space on May 1st, 2015 after a one-year preparation phase operating in a significantly smaller space. Serving as prototype, the precursor "Proto Hub" was seen as offering the opportunity to test ideas and assumptions regarding the creation of a work environment that successfully enables collaboration. This is a crucial element within the organizational concept of the Impact Hub Association. Within the founding team of Impact Hub Berlin, collaboration is regarded as the "basis to rethink economy" (Lässer) and given intrinsic importance as a work ethic in that "no decision is made alone" (Reiner).

4.2.1 Strategic Intent

In order to "shape the future of work life in general" (Kapretz) and "develop the sustainable economy of tomorrow" (Lässer), Impact Hub Berlin intends to become the center for social entrepreneurship in Germany. As such, Impact Hub Berlin functions as "a place where change-makers find a home to make ideas happen" (Reiner), based on its mission to foster "collaboration, trust and courage" (Kapretz).

In line with the Impact Hub Association's three core elements, physical space plays an important role at Impact Hub Berlin. It is perceived as the "first step to develop further" (Lässer), an "anchor point" (Kapretz), and as the "body language of the organization" (Reiner). As such, physical space serves "flexibility and multifunctionality, providing inspiration, and enabling collaboration" (founding team, Impact Hub Berlin). Flexibility and freedom of physical movement create opportunities for interaction and thereby foster innovation:

> We want a space where you can move, where you can develop a feeling of ownership, where you feel safe—to connect easily, think creatively and create new space. [. . .] We do not want a space that provokes the impression of being too perfect to move and change things—we strongly have to set the example ourselves. (Lässer)

Furthermore, to foster a culture of collective action, the organizational structure is generally non-hierarchical and promotes equal treatment for all. Subsequently, “there is no separate office for the Impact Hub team” (Kapretz) and “all the employees receive the same salary as [me], while having the same responsibility, at least concerning the workload” (Reiner).

4.2.2 Process of Creation

For Impact Hub Berlin another crucial factor in successful collaboration and innovation is learning. According to Reiner, the employees and co-workers at the Impact Hub:

> Embrace failure and learn from it. The best example is the ‘Proto-Hub’. Furthermore, the founders see themselves as leaders, not as managers and trust in the expertise of others. If our intern has invested 2 days of working time for a project, I trust his ideas and reasoning. (Reiner)

The new workspace of Impact Hub Berlin was designed applying a user-centered and iterative approach. The community was engaged in the co-creation process consisting of a number of sessions providing the opportunity to test the ideas within several feedback loops (see Fig. 4).

The co-creation process was twofold. To ensure the user-centeredness of the outcome, the community was first asked to co-design the general interior structures of the space. Architects were involved as experts to translate the wishes and needs expressed by the community in a spatial design and to ensure the feasibility of their ideas. The community then gave feedback evaluating the ideas generated by the architects. Additionally, they also added their own wishes and needs. The second part of the co-creation process was the co-construction of the actual space, including the distribution of practical tasks among members of the organization. During this phase, the overall plan of the interior design was set, while details could still be adjusted. The founders’ intention was to create a feeling of ownership among the

Fig. 4 Participants during the creation process

community, triggering a thought such as: “I made this; it was my idea and I want to maintain it well” (Kapretz et al. 2015).

The founding team conducted extensive field research, with visits to different innovation and co-working spaces, including other Impact Hubs across Europe. The team succeeded in gaining an in-depth and shared understanding of crucial components of innovation spaces. This knowledge enabled them to filter the generated ideas according to basic parameters. They prioritized the large amount of ideas and identified those that appeared to be most useful in relation to their business model. However, they stated that their selection was also influenced by personal preferences regarding style and taste.

Reflecting on the outcome, it was stated that generated ideas were not necessarily different from those of the founding team. Yet, overall the co-creation process was perceived as a valuable tool to identify important aspects, contextualize ideas and validate assumptions. “We had 100 ideas, and those 20 that correlated with those of the community seemed to be the most important ones” (Kapretz et al. 2015). Furthermore, it was said that “co-creation is not a process carried out to end up with a cool space, but rather the starting point of a process that continues/is ongoing (. . .) it doesn’t stop when the doors open” (Kapretz et al. 2015). Another aspect mentioned was the potential of this approach, especially when repeated in several sessions, it creates a momentum and cohesion that—like a snowball-effect—attracts and retains attention of people over time. The founding team plans to involve the community also in future decisions: “Whenever there is something new coming up, or in case we want to change anything, we consult our community for agreement” (Kapretz et al. 2015). In order to actively encourage the community to take part in the further development of Impact Hub Berlin, regular “Town Hall” meetings are planned. These meetings will provide the opportunity to discuss wishes and suggestions concerning the space (Kapretz et al. 2015).

In general we identified a strong motivation to involve the community in all decisions within the Impact Hub. “We never plan anything remotely and then present it to our community top down. We do everything together with our community” (Lässer 2015). This approach is also important when it comes to the question of facilitating the use of the space. “The space is built together with the founding community, therefore they don’t need any instructions or facilitation but move around naturally and freely.” Hence, the founding community serves as role models for those who join later on. The host also takes a crucial role in facilitating collaboration: the host is “the connector” (Lässer 2015). “The kitchen and chill-out area act as non-human hosts” (ibid.).

4.2.3 The Workspace Design

The new workspace at Impact Hub Berlin can be seen in regard to its physical as well as non-physical structure. In line with our conceptual approach, identifying the designed structures of the workspace provides insight into the potential of the workspace as created to foster innovation activities. Based on the defined strategic

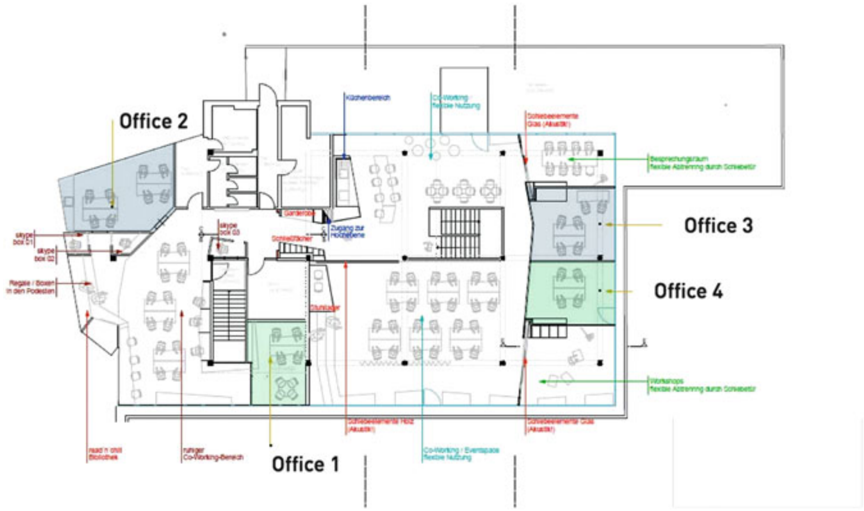


Fig. 5 Layout of Impact Hub Berlin

intent of Impact Hub Berlin, we focus on identifying spatial structures to foster collaboration as driver for innovation.

Physical features include the building's given structure, the interior architecture, as well as the atmosphere and style created (see Figs. 5 and 6).

Further physical features include the resources provided, as well as the division of the space aimed at facilitating different work modes, including different physical set-ups. Regarding its location, the building occupies a prominent position close to the city center, thus offering easy access and proximity. This potentially facilitates collaboration with surrounding institutions and incoming visitors. Resources are provided to the community in the form of office supplies and kitchen utilities. As informal meeting points such facilities promote collaboration as well. The new location is much larger than the previous "Proto Hub" and provides enough space for different zones corresponding to various work modes and preferences.

As such, Impact Hub Berlin offers space for individuals as well as for teamwork and workshops (see Figs. 7 and 8). The style is rather neutral and bright, with lots of daylight and a natural appearance, underscored by wooden pieces. An open and welcoming atmosphere is fostered, in which flexibility also plays a crucial role. Tables and chairs are easy to move around and large wooden dividers are available to create smaller workspaces if needed.

The intended function of designed physical structures is strengthened and supported by non-physical aspects. These refer to the socio-organizational work environment, program and events, with the institutionalized host acting as facilitator. Regarding the organization of space, the different zones within the hub are named according to their designated use as: event space, workshop space, team offices 1–4, meeting room, dynamic and focused working area. However, although

Fig. 6 The workspace at Impact Hub Berlin: entrance area

Fig. 7 The workspace: co-working space and the main hall

Fig. 8 The workspace: individual and teamwork facilitation areas

these names provide insight into the intended function of each work zone, the team is aware that the actual use might differ: "the names are rather functional—but we have to see if the spaces are used accordingly" (founding team—Impact Hub Berlin).

Other non-physical aspects communicate the organizational culture of Impact Hub Berlin. These include the organization's flat hierarchy, the transparency of decision-making processes and the open access to everyone interested in visiting

Impact Hub Berlin. Concerning facilitation, the art of hosting plays an important role at Impact Hub Berlin, as well as within the global institution. The host not only welcomes visitors and responds to concerns and questions, but also fosters collaboration by connecting members and arranging regular events. The host is institutionalized by Impact Hub Global. The three core elements "connect participants and are brought to life through the art of hosting." (Impact Hub Global). A host thus serves beyond the role of a receptionist, but (even more) connects people, operates as catalyzer for meaningful interactions between members, and organizes events to strengthen the community (Impact Hub Global).

Table 1 summarizes the identified key insights. We distinguish between the physical and the non-physical work-environment by using our three dimensional conceptual framework with the components of space acting as facilitator, provider, and communicator.

5 Learning from Practice

This case illustrates that workspaces designed for collaboration have a high potential to foster innovation. Besides physical structures, those that are socio-organizational are highly relevant concerning the facilitation of collaboration. Regarding the workspace as an interim stage and part of the creation process supports the objective to foster collaboration. Overall, we hypothesize that the identified organizational structures, as established at Impact Hub Berlin have a high potential to inform collaborative action.

It is interesting to notice that the management team of Impact Hub Berlin revealed a shared understanding of strategic intent, even though each of them highlighted different facets within the qualitative interviews. This commonly shared understanding of strategic intent is also prevalent in the co-creation sessions with the community and members of Impact Hub Berlin. The key imperative of "shared ownership" recurs for strategic intent, process of co-creation, and workspace design.

As this theoretical outline has demonstrated, a variety of approaches conceptualizing space co-exist. We do not evaluate nor judge either the dominance or importance of one of these theories over the other. However, in accordance with our case study, it appears that Löw's (2007) and Kornberger and Clegg's (2004) conceptualization of space provides the most suitable approach to frame our findings due to its holistic, comprehensive and relational perspective. As revealed by empirical evidence following the establishment of a clear strategic intent, the process of creating Impact Hub Berlin already incorporated traces of the reciprocity of structure and action, a stage earlier introduced and first put forward by Löw (2007).

Table 1 Impact Hub Berlin—Designed Structures to amplify collaborative culture

Physical work environment	
Facilitator	Location • Close proximity to other companies and organizations • District not fully developed, potential
	Furniture • Large wooden tables that are easy to push aside and move around • Ergonomic chairs on wheels to move easily • Focused areas supported by wooden dividers/cubicles
	Work zones/Work modes: • *Entrance/Host* (welcome, connect, introduce to culture) • *Focused Area* for teams & individuals • *Loud Collaboration* for individuals and teams • *Workshop* for collaboration/teamsprints/prototyping/exploring • *Nap Area/Terrace/*Kitchen for retreat and relaxation • *Kitchen/Terrace* for (informal) interaction/communication • *Event Space* for presentation, workshops, learning, sharing • *Meeting Space* for collaboration, sharing, presenting
Provider	Location • Prominent location in the center of the city, good infrastructure
	Architecture • Ca. 600 m^2 space • Large open space and smaller team offices
	Resources • Kitchen • Workshop space (incl. prototyping material) • Office supplies and facilities (printer, WiFi etc.)
Communicator	• Large window fronts and outdoor space • Style & atmosphere • Neutral, flexible, clear, bright • Open, welcoming, at home feeling
Social-organizational work environment	
Facilitator	Host • Welcomes visitors and new members • Shows and explains the space and its functions • Connects members to foster collaboration • Provides assistance for any concern coming from the community • Arranges member events on regular basis
Provider	Program & Events for Learning • Community events on a regular basis • Workshops • Networking events etc.
Communicator	Organizational Structures • Attention given to maintaining a balanced community as reflected in member selection • No assigned workplace, freedom to move around, self-organized space • Open access to everyone • Part of global network • Non-hierarchical • Transparent decision-making • Participative management • Co-creation

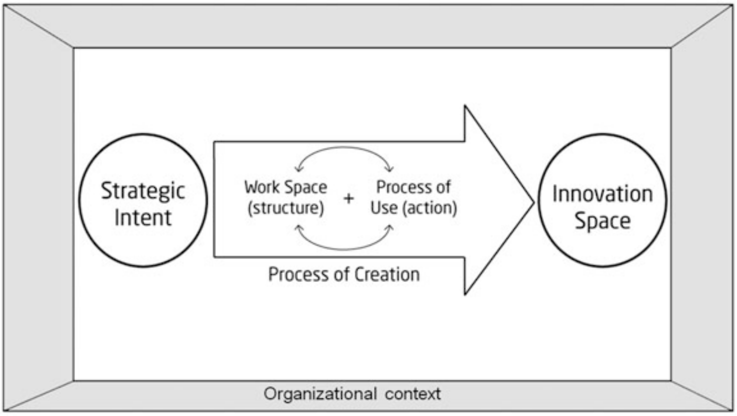


Fig. 9 Innovation space creation at Impact Hub [based on Moultrie et al. (2007)]

Furthermore, bearing in mind Kornberger and Clegg's (2004) concept of innovation space as generative building, a collaborative approach is taken with the afore-mentioned imperative of "shared ownership." Thereby, new workspaces are created as intended and the potential to turn the users into "illegal architects" of the work space by allowing them to "ab-use" the designed structures is shown. We thus argue for an iterated model of Moultrie et al. (2007) as shown in Fig. 9. The refined model focuses on the interaction of structure (workspace) and action (the process of use). The observed reciprocity is moved into focus of the process of creation. Thus, the established innovation space is the outcome of this process not an interim stage.

6 Concluding Remarks

Taking Moultrie's et al. (2007) model as processual guidance, we revealed that Impact Hub Berlin's procedure to create a workspace fostering innovation through collaboration bears promising potential. In this process we further deciphered and formulated an iterated version of the Moultrie et al. model, combining elements of Löw's (2007) conceptualization of space and Moultrie's et al. (2007) approach. We revisited the understanding of the process of creation of space by adding the reciprocity of structure and action. However, as this was a data-driven finding, our contribution has limitations. The iterated model applies and is closely related to the case of Impact Hub Berlin, whose findings can hardly be transferred to other cases without imperatives.

Nevertheless, as this study also sought to investigate how organizational leaders and changemakers can overcome imitating external best-practice examples meanwhile neglecting their company's own structure and culture, when creating innovation spaces, the research attained further general results. Impact Hub Berlin's stance on facilitating the creation of workspace internally by accounting for its specific needs (of the organization and employees) demonstrates the importance of choosing the right procedure and process to successfully create novel innovative workspaces. More importantly this approach is completely in line with the organizational culture that is driven by collective action and its subsequent impact. Therefore the common theme of "shared ownership" is the key driver for the creation of (new) innovation spaces. Such findings can be applied to other cases and contexts providing the basis to frame the strategic process of innovation space creation within existing organizational culture.

Furthermore the case study of Impact Hub Berlin also reveals that besides the deliberate model of strategic planning reflected in the initial model developed by Moultrie et al. (2007), emerging strategies exist. This shows two ends of a continuum along which innovation space-strategies lie (Mintzberg and Waters 1985).

7 Further Perspectives

The initial Moultrie et al. (2007) model visualizes a coherent understanding of the creation of space in its entirety, comprising three major steps: initial strategic intent, interim space creation and the final realized intent. It follows a deliberate model of strategy. The Impact Hub Berlin's example shows us the other end of the strategic continuum—an emergent model of strategizing (Eden and Ackermann 2013). Due to the recent opening of the space at the time of writing this article, the performance itself still needs to be assessed. For this aim, the further development of Impact Hub Berlin will be closely accompanied and regularly evaluated. An extension of our analytical framework would incorporate Moultrie's approach on process of use and realized intent complemented with further supporting data derived from the cases studied. Furthermore, it could arguably enlarge our theoretical implications by applying Lefebvre's three dimensions of space: the perceived, conceived and lived space level. In this context, comparable cases should be studied in order to gain a deeper understanding and insight into the field of collaborative workspaces. Another aspect to take into account is testing whether a similar concept of the workspace, and especially the process of creation, would be equally feasible and/or successful in a profit-oriented environment.

References

Amabile TM (1983) The social psychology of creativity: a componential conceptualization. J Pers Soc Psychol 45(2):357–376

Amabile TM (1988) A model of creativity and innovation in organizations. Res Organ Behav 10 (1):123–167

Amabile TM, Gryskiewicz ND (1989) The creative environment scale: work environment inventory. Creat Res J 2:231–252

Amabile TM, Conti R, Coon H, Lazenby J, Herron M (1996) Assessing the work environment for creativity. Acad Manage J 39:1154–1184

Anderson N, De Dreu CKW, Nijstad BA (2004) The routinization of innovation research: a constructively critical review of the state-of-the-science. J Organ Behav 25:147–173

Dark Horse Innovation (2014) Thank god it's monday. Ullstein Buchverlage GmbH, Berlin

Deb S, Sinha S (2011) Spatial influence on organizational creativity. Through syntactic analysis of space. Asian J Manag Res 1:114–129

Dul J, Ceylan C, Jaspers F (2011) Knowledge workers' creativity and the role of the physical work environment. Hum Resour Manage 50(6):715–734

Eden C, Ackermann F (2013) Making strategy: the journey of strategic management. Sage, Thousand Oaks

Friedman R (2014) The best place to work: the art and science of creating an extraordinary workplace. Perigee Books, New York

George JM, Zhou J (2002) Understanding when bad moods foster creativity and good ones don't: the role of context and clarity of feelings. J Appl Psychol 87:687–697

Guenzel S (2008) Raumwissenschaften. Suhrkamp, Frankfurt

Hamel G, Prahalad CK (1989, 2005) Strategic intent. Harv Bus Rev 83(7):148–161

IBM (2010) Capitalizing on complexity. Insights from the global chief executive officer study. IBM Global Business Services, Somers

Kornberger M, Clegg S (2004) Bringing space back in organizing the generative building. Organ Stud 25(7):1095–1114

Lefebvre H (1974) La production de l'espace (The production of space). Anthropos, Paris

Löw M (2007) Raumsoziologie (Sociology of space). Suhrkamp Taschenbuch Wissenschaft, Frankfurt/Main

Mayring P (2008) Methoden der empirischen Sozialforschung. Oldenbourg Wissenschaftsverlag, München

Mintzberg H, Waters JA (1985) Of strategies, deliberate and emergent. Strateg Manag J 6 (3):257–272

Mootee I (2013) Design thinking for strategic innovation. What they can't teach you at business or design school. Wiley, Hoboken

Moultrie J, Nilsson M, Dissel M, Haner U, Janssen S, Van der Lugt R (2007) Innovation spaces: towards a framework for understanding the role of the physical environment in innovation. Creat Innov Manag 16(1):53–65

Peters T (1992) Liberation management: necessary disorganization for the nanosecond nineties. Macmillan, London

Schensul SL, Schensul JJ, LeCompte MD (1999) Essential ethnographic methods. The ethnographer's toolkit. Altamira Press, Walnut Creek

Schepers P, van den Berg PT (2007) Social factors of work-environment creativity. J Bus Psychol 21(3):407–428

Stein MI (1974) Stimulating creativity, vol 1: individual procedures. Academic, New York

Tierney P, Farmer SM (2002) Creative self-efficacy: potential antecedents and relationship to creative performance. Acad Manage J 45:1137–1148

von Weizsäcker CF (1990) Zum Weltbild der Physik. Hirzel, Stuttgart (original 1943)

Weick KE (1995) Sensemaking in organizations. Sage, Thousand Oaks

Weinberg U, Nicolai C, Panayotova D, Klooker M (2014) The impact of space on innovation teams. In: DMI (ed) Design management in an era of disruption, London, pp 902–923

West M, Sacramento C (2012) Creativity and innovation: the role of team and organizational climate. In: Mumford M (ed) Handbook of organizational creativity. Academic, London, pp 359–385

Woodman R, Sawyer J, Griffin R (1993) Toward a theory of organizational creativity. Acad Manage Rev 18:293–321

Interviews (Case Study)

Kapretz N (2015) Personal interview, conducted by the authors, Berlin, 15 Apr 2015

Kapretz N, Lässer A, Reiner L (Founding Team Berlin) (2015) Personal group interview, conducted by author, Berlin, 15 Apr 2015

Lässer A (2015) Personal interview, conducted by the authors, Berlin, 15 Apr 2015

No Author (2015a) Impact hub global. http://www.impacthub.net/. Accessed 18 May 2015

No Author (2015b) Impact hub Berlin. http://www.impacthubberlin.net/. Accessed 18 May 2015

Reiner L (2015) Personal interview, conducted by the authors, Berlin, 15 Apr 2015

Part II
Exploring Human-Technology Interaction

Design Thinking in Health IT Systems Engineering: The Role of Wearable Mobile Computing for Distributed Care

Lauren Aquino Shluzas, Gabriel Aldaz, David Pickham, and Larry Leifer

Abstract This research examines the capabilities and boundaries of a hands-free mobile augmented reality (AR) system for distributed healthcare. We use a developer version of the Google Glass™ head-mounted display (HMD) to develop software applications to enable remote connectivity in the healthcare field, and to characterize system usage, data integration, and data visualization capabilities.

In this chapter, we summarize findings from the assessment of the SnapCap System for chronic wound photography. Through leveraging the sensor capabilities of Google Glass, SnapCap enables hands-free digital image capture, and the tagging and transfer of images to a patient's electronic medical record (EMR). In a pilot study with wound care nurses at Stanford Hospital ($n = 16$), we examined feature preferences for hands-free digital image capture and documentation; and compared SnapCap to the state of the art in digital wound care photography—the iphone-based Epic Haiku application.

The results of this study (1) illustrate the application of design thinking for healthcare delivery involving mobile wearable computing technology for distributed care, (2) improves our understanding of the benefits of human augmentation through enhanced visualization capabilities, and (3) explores a system's ability to influence behavior change through equipping clinicians with tools to improve complex problem solving and clinical decision-making in context-dependent medical scenarios. The work contributes to the future implementation of new features aimed at enhancing the documentation and assessment of chronic wounds, and provides insight into the need for future IT systems engineering projects aimed at enhancing healthcare connectivity for distributed care.

L.A. Shluzas, Ph.D. (✉) • G. Aldaz, Ph.D. • D. Pickham, R.N., Ph.D.
Center for Design Research, Building 560, 424 Panama Mall, Stanford, CA 94305-2232, USA
e-mail: lauren.aquino@stanford.edu

L. Leifer, Ph.D.
General Medical Disciplines, Stanford Medicine, 301 Ravenswood Ave. Office I238, Menlo Park, CA 94025, USA

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_7

1 Introduction

A collaborative approach to complex problem solving that leverages multiple points of view is fundamental to design thinking. In the healthcare field, the need for improved tools to enhance collaboration among patients and providers has become increasingly urgent—due, in large part, to a global rise in aging populations and chronic disease prevalence, coupled with increasing health care costs and physician shortages worldwide (Magnusson et al. 2004; Mattke et al. 2010).

To address these challenges, this research explores the application of design thinking in the field of health IT systems engineering, through the use of design collaboration technology for distributed care. In particular, we examine the use of wearable mobile computing to mediate cross-boundary communication and collaboration in the healthcare field.

This research documents the use of hands-free mobile AR for distributed healthcare and clinical decision-making. We used the Google Glass head-mounted display (HMD), which we were privileged to obtain through the Google Explorers Program. The Glass technology provides a platform for studying distributed design thinking in the healthcare field.

1.1 Prior Work

To narrow our research focus, we began needs finding at Stanford Health Care from October to December 2013. We interviewed and shadowed 16 clinicians to generate over 135 needs. We grouped needs into 15 broad clinical areas and ranked each category on a 5-point scale based on degree of importance to the hospital (pain point), alignment with research interests, and feasibility. Within the top ranking categories, we segmented needs by degree of clinical risk to patients. Our top three needs (in order of low to high patient risk) included: wound and skin care photography, point-of-view sharing during surgery, and vital sign communication during cardiac arrest (Aldaz et al. 2015; Aquino Shluzas et al. 2014). We selected chronic wound photography as a target focus area since: (1) a reduction in the incidence of chronic wounds—especially hospital-acquired pressure ulcers—is of paramount concern to healthcare facilities; (2) chronic wound image capture involves a relatively low degree of clinical risk for patients, thus enabling a solution to be tested and implemented quickly.

From a development perspective, we were interested in examining the usability features of a head-mounted display and the degree of clinical effectiveness that such a technology affords. We were also interested in examining the efficiency, satisfaction level, benefits and challenges associated with hands-free versus hands-on image capturing systems. We hypothesized that clinicians could successfully achieve their objective of hands-free digital photography through the use of a head-mounted display, controlled by gestural and voice-based commands; and that such a system could efficiently capture and document wound images, and transmit clinical data to a patient's EMR.

2 SnapCap System Design

The SnapCap System includes both a Google Glass application (known as "Glassware") and an Android smartphone application initially designed to be used by nurses for chronic wound photography (Aldaz et al. 2015; Shluzas et al. 2015). Google Glass was selected as an initial platform for research, since it has an optical head-mounted display that is capable of taking pictures and recording videos using an integrated camera. The device also has the ability to communicate wirelessly via WiFi and Bluetooth. Additional sensors integrated into the device enable the development of augmented reality-based applications. We developed the SnapCap Glassware using the Glass Development Kit (GDK) for Android 4.0.4 (API Level 15, Ice Cream Sandwich), and the smartphone app for Android 4.3 (API Level 18, Jelly Bean).

In the current implementation, an Android smartphone serves as a hypothetical EMR that includes a medical database for six fictional patients. The smartphone application communicates with Glass via Bluetooth. Presently, the Bluetooth link is unidirectional, from Glass to the smartphone application, because the app needs to store the images taken with the SnapCap Glassware as well as the associated tags. Speech-to-text conversion takes place on Google servers.

The present SnapCap architecture is for prototyping use only. Hospital implementation for routine clinical care would require encrypted communication channels and an interface with proprietary data storage to ensure privacy and security. Ultimately, we intend SnapCap to be platform-agnostic and compatible with a range of head mounted displays and EMR systems. (Complete details regarding the SnapCap system architecture are described in Aldaz et al. 2015.)

2.1 SnapCap Smartphone Application

The SnapCap smartphone application implements relevant features of the EMR, including a patient list, access to a patient database, and image storage in a media file. The password-protected application allows a clinician to select the name of the relevant patient prior to each patient encounter. The clinician can choose to either (1) take a clinical image, whereby the application establishes Bluetooth communication with Google Glass, or (2) view the patient's media file.

After choosing to take a clinical image, the nurse is free to place the Android smartphone in his or her pocket and put on Google Glass for the duration of the patient encounter. The photographs taken using Glass, along with the associated tags and voice annotations, immediately transfer via Bluetooth to the smartphone and are stored in the patient's media file, where they are available for viewing at any time.

2.2 SnapCap Glassware

The novel features of the SnapCap Glassware include (1) barcode scanning using the Glass camera, and tagging subsequent images with a patient's personal identification information that is embedded in the barcode, (2) capturing a live video preview in the Glass eyepiece before a photo is taken, (3) using a double blinking gesture to take photographs, through utilizing Glass' IR sensor, and (4) using a head tilt gesture (while in the preview mode) to zoom in and out of an image, and a head title gesture to send images to a patient's EMR, through the use of Glass' internal measurement unit (IMU) sensor.

3 Initial Usage and Feasibility Assessment

As an integral part of system development, in order to better understand how SnapCap might serve the wound documentation needs of nurses, we conducted an initial feasibility and usage assessment of the SnapCap System with five wound care nurses and two physicians at Stanford Health Care ($n = 7$ participants) (Aldaz et al. 2015). Each assessment consisted of two parts and lasted approximately one hour. First, we reviewed the Google Glass navigation basics with each participant. We then asked each participant to complete a series of tasks in order for the research team to qualitatively evaluate the operation of Google Glass, and users' preferences toward new and existing Glass features. The tasks included:

1. Using voice and touch-based commands to take a picture and record a video, using Glass's standard features (without image preview and zooming).
2. Evaluating the use of a customized camera zoom application, with image preview, that allowed users to zoom in and out through blinking one or both eyes in rapid succession.
3. Assessing the use of the customized camera zoom application with image preview, to zoom in and out through a one-finger gesture (sliding one's finger back and forth on the Glass touch pad).
4. Evaluating the use of Glass's existing historical image retrieval feature, in which users scroll forward on the Glass touchpad to see a list of previously taken images.

In the second part of the assessment, we asked clinicians to answer six, multi-part questions regarding their preferred methods for capturing, cropping, annotating and retrieving wound images, along with considerations for sterile image capture techniques.

From this qualitative assessment, we learned that mobile, hands-free operation was critical for wound image capture and annotation. None of the clinicians wanted to touch a head-mounted image capture system with potentially contaminated gloves. We also learned that, as a baseline functionality requirement, the new system must perform as well as the current Epic Haiku digital image application, which is the current state of the art in wound photography at Stanford Health Care (Aldaz et al. 2015). Based on lessons learned from the initial feasibility assessment, we revised the SnapCap Glassware design with modifications and customized features, to address users' needs before conducting the wound care pilot study.

4 Pilot Study: Image Capture and Documentation

We conducted a two-part within-subjects lab-based pilot study with 16 nurses from Stanford Health Care (15 female, 1 male; 11.7 ± 8.7 years of wound care experience). The first part focused on an evaluation of core features of the SnapCap System and a comparison between the use of SnapCap and the current state of the art in digital wound care photography, the iPhone-based Epic Haiku application. The second part of the pilot study involved collecting and analyzing speech-to-text data for wound annotations (Aldaz et al. 2015).

We conducted a lab-based pilot study with hypothetical patients in order to focus on user interaction preferences for digital image capture and documentation, and to ensure that each nurse captured the same images, as a means of direct comparison between the two applications for wound photography. Study participants were recruited from a group of wound and ostomy care nurses at Stanford Health Care. None of the nurses had prior Google Glass experience. All 16 nurses had prior smartphone experience, while three nurses had worked directly with Epic Haiku.

4.1 Part 1: SnapCap Feature Evaluation and Application Comparison

For part one of the evaluation, we brought each nurse into a hospital room where we had placed two pelvis-only mannequins and an identifying barcode on a table. The mannequins are normally used to train students in the assessment and treatment of pressure ulcer wounds. A researcher explained the purpose of the study, the time breakdown per user session, the Glass navigation basics, and the experimental set-up. A researcher explained the use of both SnapCap and Epic Haiku, and asked each nurse to photograph a wound on a mannequin using (1) SnapCap (running on Glass and a Galaxy Nexus smartphone) and (2) Epic Haiku (running on an iPhone 4S). The photographs taken by nurses (using SnapCap) were saved in the patient's hypothetical EMR in order to examine the quality of each photo taken.

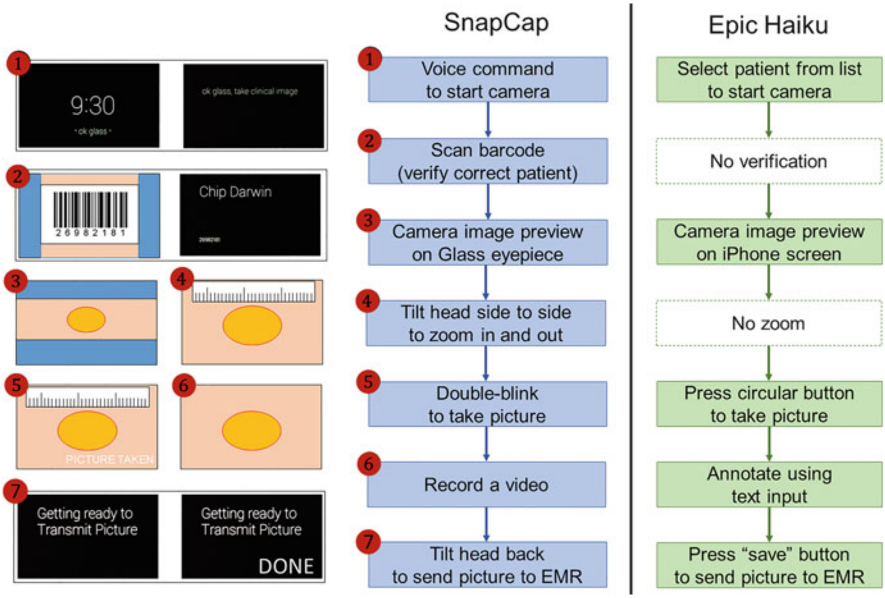


Fig. 1 Wound photography flow for SnapCap (*left*) and Epic Haiku (*right*) (Aldaz et al. 2015)

For wound annotation, each nurse was asked to use text-based documentation in Epic (per standard documentation practices) and to annotate the wound in a 10-s video recording using the SnapCap system. Figure 1 shows the step-by-step flow in each session. The research team took manual notes and recorded each session using a digital recording device. Photographs and brief videos of nurses using the SnapCap System and Epic Haiku application were also captured. Each user session, consisting of the SnapCap evaluation, application comparison, and a post-task questionnaire, lasted 30–45 min per participant.

4.2 Part 1: Post-task Questionnaire

Following the use of each system, participants completed a pen-and-paper questionnaire that was segmented into five areas. The questionnaire first asked users to provide their number of years of wound care experience and smart phone experience, and to list mobile applications that they commonly use. The questionnaire then asked participants to share their preferences for (1) current SnapCap system features, (2) application preferences for SnapCap versus Epic Haiku, and (3) preferences for the implementation of future SnapCap features. The research team saved the questionnaire data in an Excel spreadsheet for data analysis.

Glass Feature Preferences In the first part of the post-task survey, participants were asked to indicate their preference for the following SnapCap features:

1. Barcode scanning for patient identification
2. Voice-based documentation via a brief video recording
3. Double blinking gesture to take photographs
4. Head tilt gesture for zooming in and out of an image

Response options were prefer, do not prefer, or recommended improvements (with space to provide improvement recommendations). For each question, a response was required and multiple response options could be selected.

Application Preferences In the second part of the post-task survey, participants were asked to indicate their preference between SnapCap and Epic Haiku for the following performance dimensions:

1. Considerations for sterile wound image capture technique
2. Photo-capture capability
3. Image quality
4. Overall Ease of Use

Response options were Epic Haiku, Google Glass, no difference, and neither. For each question, a single response was required.

Preferences for Future SnapCap Features In the third part of the post-task survey, participants were asked to indicate their preferences regarding the perceived benefit of future SnapCap features:

Response options were yes, no, and no preference. For each question, a single response was required.

4.3 Part 2: Speech-to-Text Wound Annotation

Following the initial user session, we conducted a 10–15 min follow-up session at Stanford Health Care with the original 16 nurse participants, to obtain quantitative data on the performance of Google's speech-to-text (STT) engine. In these sessions, we asked each nurse to read aloud a wound annotation, consisting of two fictitious descriptions, while wearing the Glass head-mounted display. For example:

> This is a stage 3 pressure ulcer, measuring 11 centimeters by 6 centimeters by 3.4 centimeters deep, full-thickness ulceration which probes to bone, circumferential undermining, 2.5 centimeters at 12 o'clock and 0.2 centimeters at 6 o'clock.

Results of the transcription were stored locally in Glass' built-in memory for analysis.

5 Results

For the SnapCap feature evaluation and application comparison (Part 1 of the pilot study), we used the Wilcoxon Signed-ranks test, a non-parametric statistical hypothesis test, to evaluate differences in mean ranks between two response options. The analysis was conducted using IBM SPSS statistical software. The details (data analysis and complete results) can be found in Aldaz et al. (2015).

6 SnapCap System Evaluation: Hands-Free Digital Image Capture and Documentation

We evaluated nurse preferences for features and interactions aimed at enhancing digital image capture and documentation. These included: barcode scanning, voice-based documentation through video, double blinking, and head tilt. Figure 2 illustrates the difference between the sum of ranks for feature preferences. In a follow-up study, we examined the accuracy of Google's STT engine for wound documentation.

6.1 Barcode Scanning for Patient Identification

A Wilcoxon Signed-ranks test indicated that there was a statistically significant preference for hands-free barcode scanning for routine clinical care, $Z(15) = -3.873$, $p < 0.001$, $r = 0.71$. All 16 nurses successfully used the SnapCap Glassware to read the patient barcode within 4 s. One nurse indicated that, "Google Glass has the ability to barcode patient identity to reduce errors."

6.2 Voice-Based Documentation Through Video

The data analysis indicated that voice-based documentation through a brief video recording was strongly preferred by nurses, $Z(15) = -2.84$, $p = 0.005$, $r = 0.52$. The qualitative data also indicated that nurses favored the use of voice commands for launching the video recording and image capture features of the SnapCap System. However, the data revealed that current voice-commands were challenging for participants to use. Diverse failure modes included saying an incorrect phrase (3 nurses), saying a phrase too quickly (1 nurse), or saying a phrase too softly (1 nurse). While documenting a wound through a verbally annotated video, nurses frequently commented that the 10-s duration was too short, and that additional time was required.

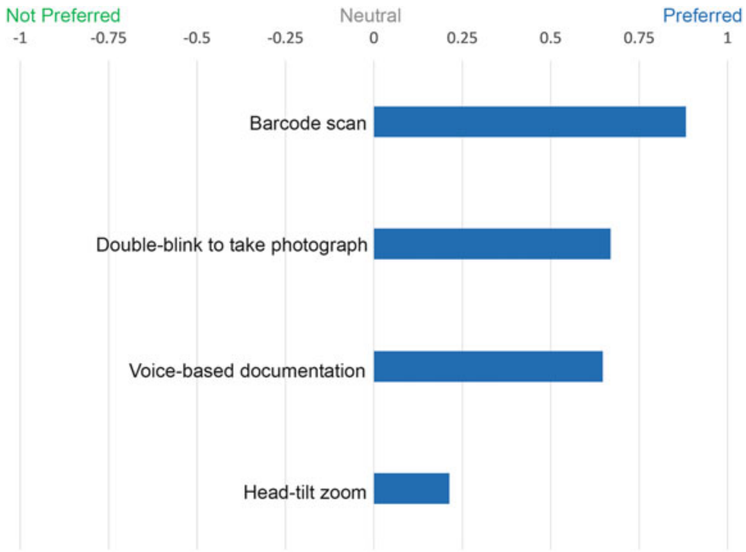


Fig. 2 Normalized difference between the sum of ranks for SnapCap feature preferences (Aldaz et al. 2015)

6.3 Double Blinking to Take Photographs

Nurses took a photograph (made the camera shutter open and close) by double blinking. Overall, double blinking was well received, with a statistically significant preference for this feature, $Z(13) = -3.606$, $p < 0.001$, $r = 0.71$. One nurse took five or six photographs inadvertently because Google Glass registered her natural eye-blinking rate as double blinking. Another cautioned that double blinking "may be too variable depending on the individual."

6.4 Head Tilt for Zooming

Nurses achieved camera zoom in and out by tilting their head to the right and left, respectively. Overall, this was the least preferred hands-free interaction method, $Z(10) = -1.897$, $p = 0.058$, $r = 0.42$. One nurse commented that the zoom was slow to respond, and three more indicated that this feature might interfere with their ability to assess wounds or pressure ulcers (for example on the back or the heel), which usually required them to tilt their heads from side to side. They felt that seeing the image zoom in and out on the eyepiece display could potentially interfere with their ability to take a high quality photograph.

6.5 Speech-to-Text Annotation

When we presented nurses with the possibility of annotating a wound through speech-to-text transcription, they were generally excited about this feature, but anticipated having to access the text for review and editing if necessary. From data gathered during the follow-up user session, we evaluated the performance of the Google speech-to-text (STT) engine for transcribing wound annotations. Due to technical issues, 2 of the 16 annotations had missing data—in each case, several words were missing from the transcript. Additionally, we noticed three instances when a nurse uttered an incorrect word, so we removed these words from the data set.

To analyze the Google speech-to-text (STT) data, captured in the follow-up user sessions, we used Word Error Rate (WER) as a standard measure for examining transcription accuracy (Jurafsky and Martin 2014). The overall WER of Google's STT for the fictitious wound description, among the 16 annotations evaluated, was 21.9 %. For individual nurses, the WER ranged from 7 to 38 % (SD = 10.3 %). From a technical standpoint, the Google STT performed well, in conjunction with the Glass hardware, in a majority of the tests. Fourteen out of sixteen (87.5 %) nurses successfully saved the two wound annotations in four attempts or less (although examination of the text files revealed a small amount of missing data on 2 of the 14 data files). Two of sixteen nurses (12.5 %), however, required more than ten attempts to successfully save the two wound annotations. In both cases, the problem with saving annotations were due to poor network connectivity, which caused the program to hang-up in mid-transcription (Aldaz et al. 2015).

7 Comparison Between SnapCap and Epic Haiku

We compared the SnapCap System and Epic Haiku for digital wound photography. Four aspects we investigated were: (1) sterile image capture technique, (2) photo capture capability, (3) image quality, and (4) overall ease of use. Figure 3 shows the difference between the sum of ranks for application preferences.

7.1 Sterile Image Capture Technique

A Wilcoxon Signed-ranks test indicated that there was a statistically significant preference for the Glass SnapCap System versus Epic Haiku in regard to sterile image capture technique when photographing wounds, $Z(16) = -3.873$, $p < 0.001$, $r = 0.68$. Comments relating to the SnapCap system included: "Much better experience," and "Not keen on touching the Glass frame with gloved hands during a consult."

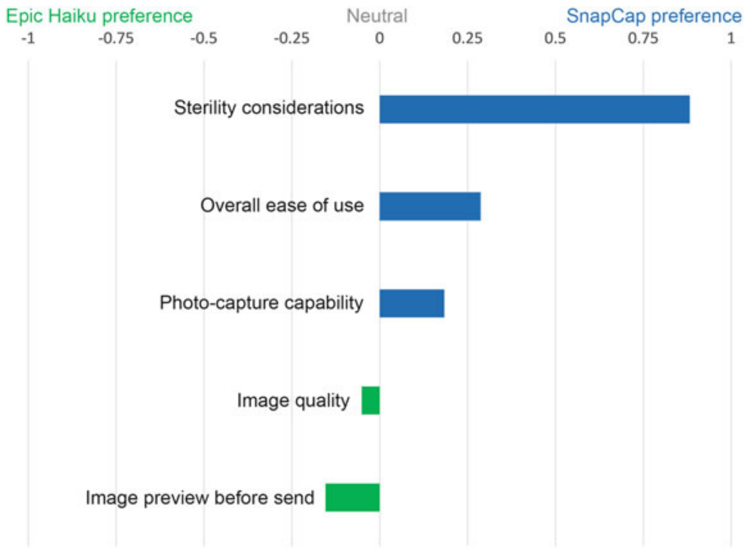


Fig. 3 Normalized difference between the sum of ranks for application preferences (Aldaz et al. 2015)

7.2 Photo Capture Capability

The data analysis revealed that there was no significant difference in preferences between the Glass SnapCap system and Epic Haiku for photo capture capability, $Z(16) = -1.667$, $p = 0.096$, $r = 0.29$. Although, qualitative data illustrate that nurses preferred to preview images on a relatively large iPhone screen, as opposed to the smaller Glass display. One nurse commented, "Google Glass has the ability to barcode patient identity to reduce errors, but needs better pixel definition."

7.3 Image Quality

There was no significant difference in preferences between the image quality of the SnapCap System and Epic Haiku, $Z(15) = -0.816$, $p = 0.414$, $r = 0.15$. Nurses agreed that the quality of photographs taken should be sufficient for making clinical decisions, and qualitatively perceived image quality as a weakness of SnapCap. Using SnapCap, the 16 nurses transmitted 30 wound photographs from Glass to the smartphone app, of which we considered 17 of 30 (56.7 %) to be of acceptable quality for clinical use. Challenges included blurred (7/30, 23.3 %), tilted (3/30, 10 %), and improperly framed (3/30, 10 %) photos.

7.4 Overall Ease of Use

The data illustrate no significant difference in preferences between the SnapCap system and Epic Haiku in terms of ease of use, $Z(14) = -1.732$, $p = 0.083$, $r = 0.33$. Given the lack of familiarity that the nurses had with head-mounted displays, we see the limited difference in ease of use preferences as an encouraging result. One nurse commented that, "Epic Haiku is more 'normal' for today's registered nurses, but Google Glass has some great possibilities." Another nurse called using SnapCap for wound photography, "a much better experience" then Epic Haiku.

8 Preferences for Future Image and Data Sharing Features

The survey data captured user preferences regarding potential new features for improved image and data sharing, for digital wound care photography. These include the use of a head-mounted display (HMD) for bi-directional communication with colleagues, features for historical image retrieval for data recall and sharing, and the use of a dynamic digital ruler inside the eyepiece of an HMD.

8.1 Use of a Head-Mounted Display to Share and Discuss Images

A Wilcoxon Signed-ranks test indicated that there was a statistically significant preference for the future use of head-mounted displays for sharing and discussing wound images among colleagues, in order to obtain real-time feedback on a diagnosis or to aid in clinical decision-making, $Z(13) = -3.606$, $p < 0.001$, $r = 0.71$. One nurse commented that she felt such a system was, "an interesting idea" and another mentioned, "it is a definite possibility."

8.2 Historical Image Retrieval for Data Recall and Sharing

The data illustrate a significant preference for historical image retrieval, to achieve time-lapse image recall in a head-mounted display after taking a series of photographs, $Z(14) = -3.742$, $p < 0.001$, $r = 0.71$. Suggested uses for this feature included the ability to see changes in wound margins over time, and to track a wound's staging and healing progression (for stage 4 to stage 1 pressure ulcers). One nurse commented that it "would show progression or deterioration of the wound." Another mentioned, "this is very important for staging pressure ulcers—one must know the previous stage so one does not downstage the ulcer."

8.3 Dynamic Digital Ruler Inside the Glass Eyepiece

The data revealed a significant preference for a digital ruler inside the eyepiece of a head-mounted display, to replace a hand-held paper ruler, $Z(13) = -3.051$, $p = 0.002$, $r = 0.60$. Comments included: "Need to work on accuracy, but yes, that would be great so that you wouldn't have to hold or dispose of the ruler," and "Huge help!"

9 Conclusion

SnapCap enables hands-free digital photography, tagging, speech-to-text image annotation, and the transfer of data to an electronic medical record. In its current implementation, SnapCap leverages Google Glass' camera and internal sensors to guide clinicians through the process of taking and annotating wound images. This book chapter documents the SnapCap architecture and examines user preferences regarding hands-free (voice and gesture-based) interactions for image capture and documentation, based on a pilot study with sixteen wound care nurses at Stanford Health Care. We compare the SnapCap Glass application with the current state of the art in digital wound care photography.

The data illustrate that nurses strongly favored SnapCap's ability to rapidly identify patients through barcode scanning. They also favored the use of voice-based commands to launch applications and to document wounds, as well as the double blinking action to take photographs. However, users expressed mixed views regarding head tilt gestures for zooming while previewing images. In a head-to-head comparison with the iPhone-based Epic Haiku application, users strongly preferred the SnapCap System for sterile image capture technique when photographing wounds. Yet, preferences were divided in regard to photo capture capability, image quality, and overall ease of use. The similar ease of use scores for the SnapCap and Epic Haiku systems was promising, given the lack of prior experience that nurses had with head-mounted displays in comparison to smartphones.

In considering future research directions, this work reveals that clinicians need a hands-free, rapid, convenient, and affordable device that provides wound measurements with a reasonable level of reliability and repeatability. Such a system would help to address the growing problem of chronic wounds—which affect 6.5 million Americans and pose a $25 billion annual financial burden to the U.S. health care system. A repeated wound measurement system would be a vital component of overall chronic wound assessment (for pressure ulcers in particular), as it provides an indicator of the percent reduction or increase in wound area over time.

The research summarized in this chapter provides a foundation for the development of new integrated applications for the capture, tagging, and transfer of digital images for wound care and other clinical applications. This work illustrates the application of design thinking for healthcare delivery that involves mobile wearable computing technology for distributed care. By using the design and evaluation of technology as a platform for research, this work improves our understanding of the benefits of human augmentation through enhanced visualization capabilities. It explores a system's ability to influence behavior change through equipping clinicians with tools to improve complex problem solving and clinical decision-making in context-dependent medical scenarios.

References

Aldaz G, Shluzas LA, Pickham D, Eris O, Sadler J, Joshi S (2015) Hands-free image capture, data tagging and transfer using Google Glass: a pilot study for improved wound care management. PLoS One 10(4), e0121179. doi:10.1371/journal.pone.0121179

Aquino Shluzas L, Aldaz G, Sadler J, Joshi SS, Leifer L (2014) Mobile augmented reality for distributed healthcare: point-of-view sharing during surgery. In: Proceedings of the Fourth International Conference on Ambient Computing, Applications, Services and Technology (Ambient 2014), Rome, Italy, 24–28 August

Jurafsky D, Martin JH (2014) Speech and language processing, 2nd edn. Pearson Education, Upper Saddle River, NJ

Magnusson L, Hanson E, Borg M (2004) A literature review study of information and communication technology as a support for frail older people living at home and their family carers. Technol Disabil 16:223–235

Mattke S, Klautzer L, Mengistu T, Garnett J, Hu J, Wu H (2010) Health and well-being in the home: a global analysis of needs, expectations, and priorities for home health care technology. RAND Corporation, Santa Monica, CA

Shluzas LA et al. (2015) A wearable computer with a head mounted display for hands-free image capture, measurement, speech-to-text translation, and bi-directional communication with external sensors. US Provisional Patent No. 62134580, Mar 2015

Redesigning Medical Encounters with Tele-Board MED

Anja Perlich, Julia von Thienen, Matthias Wenzel, and Christoph Meinel

Abstract The roles and perspectives of the patient and the health care provider could hardly be more different, yet both pursue the common goal of restoring or preserving the patient's health. The path to a satisfying health care outcome is manifold, and the quality of the patient-provider relationship is an impactful factor. We discuss different models for the classification of patient-provider interaction as well as for patient empowerment. On this theoretical basis we elaborate on how patient-provider interaction can be enhanced in practice by means of the medical documentation system Tele-Board MED. This system is a collaborative eHealth application designed to support the interaction between patient and provider in clinical encounters. Simultaneously, it aims at making case documentation more efficient for providers and more valuable for patients. As a research paradigm, the Tele-Board MED project has used a design thinking approach to understand and support fundamental stakeholder needs. Psychotherapy has been chosen as a first field of application for Tele-Board MED research and interventions. This chapter shares insights and findings from empathizing with users, defining a point of view, ideating and testing prototypes. We found that a joint, transparent case documentation was very well received by patients. This documentation increased the acceptance of diagnoses and encouraged a team feeling between patient and therapist.

1 Different Perspectives on Clinical Encounters

When a person suffers from an illness and decides to see a doctor that person becomes a patient. Patients, who seek help, take the step of confiding in a healthcare professional about very sensitive issues. Most of the time there are physical or mental symptoms that force a person to see a doctor. It is not an enjoyable activity, but rather a necessary evil that interferes with everyday plans. Besides the

A. Perlich (✉) • J. von Thienen • M. Wenzel • C. Meinel
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: anja.perlich@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_8

observation of symptoms, there are other precursors of the visit to the doctor: determining what might be wrong, browsing the internet and other information sources, choosing the right practitioner, making an appointment, and so on. After the relatively short meeting with the doctor, the patient awaits the diagnosis and then ponders therapy options. This scenario is followed by a possible referral to a specialist or the procurement of medication from the pharmacy.

For doctors—we might call them providers, practitioners, health care professionals, physicians or therapists—patient consultations are daily business. Doctors are in the position of exercising their profession with the ultimate goal of preserving or restoring health. In the time before the meeting, the doctors prepare a new patient file or look into notes from previous sessions. Afterwards, as part of their legal obligation and for their own records they complete the documentation, code diagnoses and treatments to assure a correct billing.

The roles and perspectives of patient and practitioner could hardly be more different and the meeting itself significantly determines the course of the treatment and ultimately the cure of the disease.

This chapter takes a closer look at clinical encounters and the potential to enhance patient-provider interaction and patient empowerment through the use of information technology. In Sect. 2 we start with a discussion of existing models for classifying patient-provider relationships from a patient's point of view (Sect. 2.1) and from an interaction process perspective (Sect. 2.3). In this context we also elaborate the role of the internet as a central factor for increasing knowledge accessibility for patients (Sect. 2.2). In Sect. 3 we make the transition to the concept of patient empowerment and depict eHealth technologies as a central driver for the shift of power from provider to patients. We discuss the contact points of patient-provider interaction in the field of computer supported collaborative work (Sect. 3.2) and propose the medical documentation system Tele-Board MED as a tool to support collaboration in clinical encounters. Section 4 describes our research in one medical domain, namely psychotherapy, where patient-provider interactions have been studied with and without Tele-Board MED. More specifically, we present the feedback we got from patients and therapists (Sect. 4.4). Some of the striking effects we found on patients were: an increased concordance with therapeutic treatments, and the creation of the feeling that patients and their therapists were part of a team. Section 5 summarizes how Tele-Board MED contributes to patient empowerment and positive patient-provider interactions.

2 The Patient-Provider Relationship

The work of successful designers is strongly informed by the needs of the person for whom they design. This is the case whether the work takes the form of products, services or software applications. In our research project, Tele-Board MED, we aim at creating a documentation tool that supports patients and providers individually as well as in their interaction together. While we have investigated provider perspectives on Tele-Board MED previously (Perlich et al. 2014; von Thienen et al. 2016), we are now concerned with a more comprehensive understanding of the relationship between patient and provider in order to build something meaningful for both.

Even though the roles and perspectives of patient and provider are very different from each other, both parties pursue a common goal—to restore or preserve the patient's health. The path to a satisfying health care outcome is manifold, and the quality of the patient-provider relationship is an impactful factor.

During much of the twentieth century the relationship between patient and provider was a patriarchal one. Physicians had exclusive access to medical knowledge, and thus the power and responsibility of decision-making. As counterpart to the dominant position of the provider, the patient assumed the role of the obedient healthcare recipient.

When ideals in society started to change and the opportunities for patients to acquire medical knowledge increased, this hierarchical model started to become outmoded. Over the last decades the balance of power and responsibility in healthcare is shifting from care providers to patients.

2.1 Classifying Relationships According to Patient Characteristics

There is an ongoing discourse about the nature of medical encounters. Research literature yields diverse classification systems and models which comprise ethical, psychological and sociological aspects (Agarwal and Murinson 2012).

A popular classification system was proposed in 1992 by Emanuel and Emanuel. A scale with stepwise increasing patient involvement describes the patient-provider relationship as paternalistic, deliberative, interpretive or informative (Emanuel and Emanuel 1992). The two central variables of patient values and patient autonomy determine the classification of the relationship into one of these four categories (Fig. 1, left). In this context, patient values are beliefs or principles related to personal health and the medical sphere (e.g., the extent to which a person values life versus lifestyle). These values have an impact on treatment decisions and commitment to health-sustaining activities (ibid.).

Twenty years later, in 2012, the model was extended to take account of the increasing availability of medical information and technological advances. A third dimension, the patient's medical knowledge, was added by Agarwal and Murinson. With their patient-centered interaction model, they want to account for high patient diversity (Fig. 1, right).

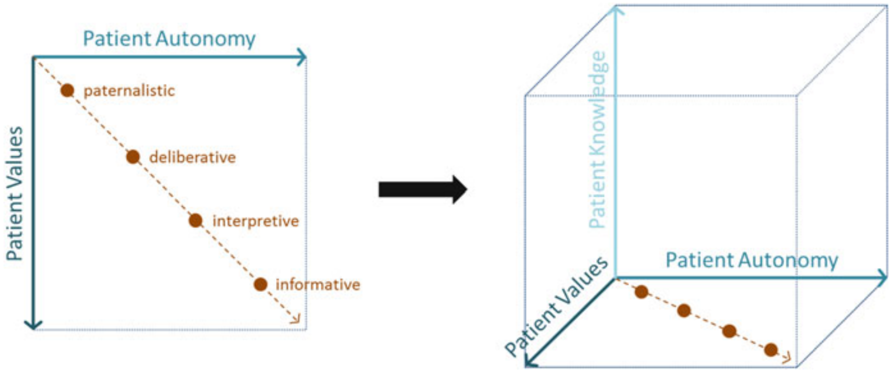


Fig. 1 Patient-centered systems for the classification of patient-provider relationships. The four-step scale presented by Emanuel and Emanuel (1992) was reworked and extended by Agarwal and Murinson (2012). The new system is a three-dimensional model where patient values, knowledge and autonomy can be assessed independently

2.2 *How the Information Revolution Changes Health Care*

In the past, the patient's level of knowledge was determined by the healthcare provider, who disclosed more or less medical knowledge and clinical case-specific information. Today patients can be better informed both in breadth and in depth. The number of medically relevant publications in print and electronic media is increasing. Above all the internet is the most important factor for the exponential growth of medical knowledge accessibility (Agarwal and Murinson 2012). The amount of health-related information that can be found is vast—whether it deals with material on symptoms, illnesses, treatments or medication. Even if detailed information on a very specific condition is the subject of a search, relevant expert articles or research publications can be found on the internet. The knowledge these sources contain might even surpass that of the patient's practitioner.

The accessibility of health information on the internet has a considerable impact on health care in general and on the patient-provider relationship in particular. In a large-scale study conducted in the USA, Murray et al. investigated the physicians' (Murray et al. 2003a) as well as the patients' (Murray et al. 2003b) perspectives on the impact of health information on the internet. The quality of the information found on the internet by patients and brought to the visit is of key importance. Correct, relevant information is beneficial, while incorrect, irrelevant information is harmful for the patient-provider relationship (ibid.).

From the point of view of physicians, skepticism prevails towards this new information source. Doctors express concern about the validity of health information found on the internet as well as their patients' ability to judge its quality sufficiently. Incorporating patient-researched information in the clinical encounter also leads to new dynamics. For instance, in order to avoid harming the patient-provider relationship or negatively affecting time efficiency, physicians tend to tolerate patients' requests that are clinically-inappropriate (Murray et al. 2003a).

For patients, on the other hand, the additional knowledge provided by the internet seems to be tremendously beneficial. Patients believe that it improves the understanding of their health problem and their decision-making ability. They also feel that looking for information helps them to take better care of their own health and gain increased confidence in speaking to their care provider.

While physicians fear that patient-researched information might be detrimental for the relationship, patients think that it leads to improved communication. They also contend that additional information encourages them in their efforts to follow their physician's advice (Murray et al. 2003b).

Koch (2012) points out the crucial role of information access in enabling patients. She says that "access to information, building knowledge, and transforming knowledge into action" are the three mandatory steps in the process of patient empowerment.

2.3 The Social Control Continuum

Interestingly, the discourse about a more active involvement of patients in health care was already going on long before the "information revolution." Already in 1978, the patient-provider relationship was seen as a variable on the social-control continuum, composed of the three models of compliance, adherence, and therapeutic alliance (Barofsky 1978). Compliance suggests that the patient is being coerced. Adherence implies that the patient is conforming to an expected standard. In contrast, alliance clearly goes in the direction of a balanced patient-provider relationship. "As an ideal patient-provider relationship, therapeutic alliance is defined as a dynamic interactional process in which the patient and provider collaborate to carry out negotiated mutual goals in a shared partnership." (Kim et al. 2008, p. 85).

The social control continuum was picked up by Kim et al. and underpinned with four dimensions of a patient-provider relationship. These dimensions are: collaboration, integration, empowerment, and communication (Kim et al. 2001, 2008) (Fig. 2). They developed the "Kim Alliance Scale" to measure the quality of therapeutic alliance, tested it with patients and refined it. The scale has four subscales that correspond to the four dimensions. Each subscale has four items, each of which describes the patient's perspective on the relationship.

The dimension of collaboration refers to negotiating shared goals for the patient's care and pursuing them. The process of integration involves mutual respect and a reduction of the power differential between patient and provider. Power is equalizing, for instance, when not only providers bring in knowledge and skills, but also when patients add specific experiences about their conditions. In practice this requires an atmosphere where the patient feels respected and encouraged to state personal opinions, which are heard without being criticized. In the

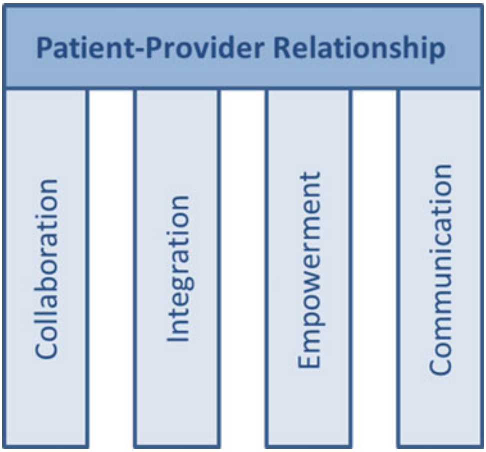


Fig. 2 Process-centered model for describing patient-provider relationships. Collaboration, integration, empowerment, and communication are the four dimensions of the interaction process, which concern both patient and provider equally [figure inspired by Kim et al. (2008)]

empowerment process, patients develop self-efficacy, take on more responsibility for their own care, and become partners in making decisions. The fourth aspect of therapeutic alliance is communication. This dimension of communication refers to a mutual understanding and information exchange, as well as patient-provider bonding. In practice, the provider supports the patient's point of view in a nonjudgmental and empathic manner, and communicates in a way that is understandable for the patient.

2.4 Combining Two Approaches: Patient-Centered Versus Process-Centered

In Sects. 2.1 and 2.3, previously described above, we examined two approaches found in the research literature that operationalize and break down the issue of patient-provider relationship. The framework for patient-physician interaction presented by Agarwal and Murinson (2012) builds on the three dimensions of: patient values, patient autonomy and patient knowledge. This approach is patient-centered in the sense that all three dimensions (values, autonomy, and knowledge) focus on patient attributes (Fig. 1).

In contrast, the patient-provider relationship framework presented by Kim et al. (2001) consists of four dimensions that describe the interaction process between patient and provider. These four pillars are collaboration, integration, empowerment, and communication (Fig. 2).

Both approaches, in combination, highly resonate with the broader concept of "patient empowerment," involving both the patient's perspective and the interaction perspective. In Fig. 3 we propose a patient empowerment model where both approaches are combined.

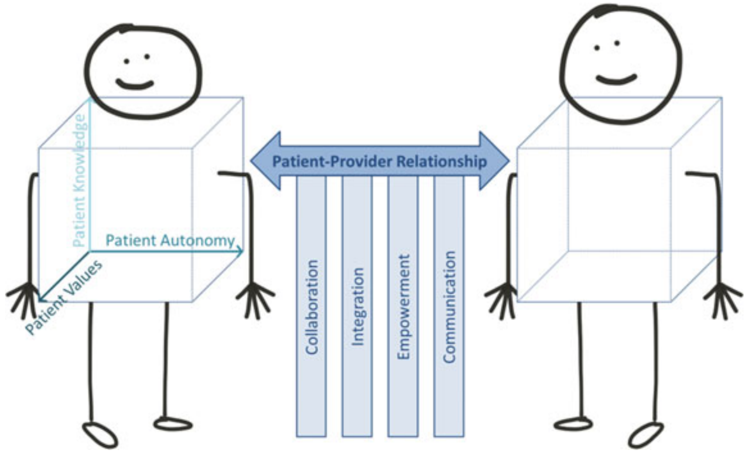


Fig. 3 The patient empowerment model is a combination of the patient attributes (cf. Agarwal and Murinson 2012) and the dimensions of patient-provider relationship (cf. Kim et al. 2008)

3 Patient Empowerment Through eHealth

The empowerment of patients is gaining considerable importance across various domains of health care and is a topic of lively discussion in research. The World Health Organization (WHO) laid the foundations for an international discourse with their definition of health promotion as "the process of enabling people to increase control over, and to improve, their health" (1986). This process can be described as having two dimensions, focusing either on the patient or on the provider-patient interaction (Aujoulat et al. 2006). The former perspective looks at the patient's personal transformation and the desired outcome to gain more power over and improve the quality of one's life. The second perspective is characterized by a communication and education process through which patient and provider collaborate and where values, power and knowledge are shared. We have come across these three aspects with a slightly different emphasis before—namely as the dimensions of the patient-centered framework by Agarwal and Murinson (2012) (Fig. 1).

Barr et al. (2015) review 25 years of patient empowerment research, systematize patient empowerment measures and present a tentative definition. According to their view, patient empowerment can be seen as "[. . .] a process achieved through patient-centered care, or as an outcome, and includes elements relating to both patient and healthcare professional roles, shared decision-making, patient self-efficacy and coping" (ibid., p. 14).

3.1 The Changing Roles of Patients and Providers, and the Emerging Role of eHealth

The healthcare sector is undergoing a transformation. The current state of healthcare requires changes, including how healthcare is delivered. The changing role of patients actively participating in the processes of their care requires that providers adjust to a new role as well. Care providers are moving away from a position of being considered pure experts who possess exclusive domain knowledge. They are moving more into the role of facilitators who enable patients to carry out informed decision making. Internet technologies, mobile devices, and in this context upcoming eHealth techniques, advance this transformation in the healthcare sector. The buzzword and umbrella term 'eHealth' generally relates to the use of information and communication technology for health and wellbeing. A special interest group of the Healthcare Information and Management Systems Society (HIMSS) defined eHealth as: "The application of Internet and other related technologies in the healthcare industry to improve the access, efficiency, effectiveness, and quality of clinical and business processes utilized by healthcare organizations, practitioners, patients, and consumers to improve the health status of patients" (2003, p. 1).

In order for an eHealth solution to support the patient empowerment process, it needs to support its three mandatory steps: "access to information, building knowledge, and transforming knowledge into action" (Koch 2012, p. 26). Just as patient empowerment definitions and patient-provider relationship models have focused on two issues (namely patient perspective versus patient-provider interaction); existing eHealth solutions also have different foci and can be classified accordingly. On the one hand there are solutions that support patients in their personal process of transformation towards becoming more active players in their care. The range of eHealth solutions designed for patients is broad and extends from internet-based patient portals and social networks to sensor-based health-enabling technologies and further to personal health records (Koch 2012). On the other hand, there are applications that are intended to support patient-provider interaction and collaboration.

3.2 Collaborative eHealth and Medical Records

eHealth for collaboration clearly overlaps with the research field of Computer Supported Cooperative Work (CSCW). This term "combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques" (Wilson 1991, p. 6). The objective of CSCW is to design computer-based technologies for cooperative work arrangements (Schmidt and Bannon 1992). The term "groupware" refers to software and/or hardware artefacts that support group cooperation.

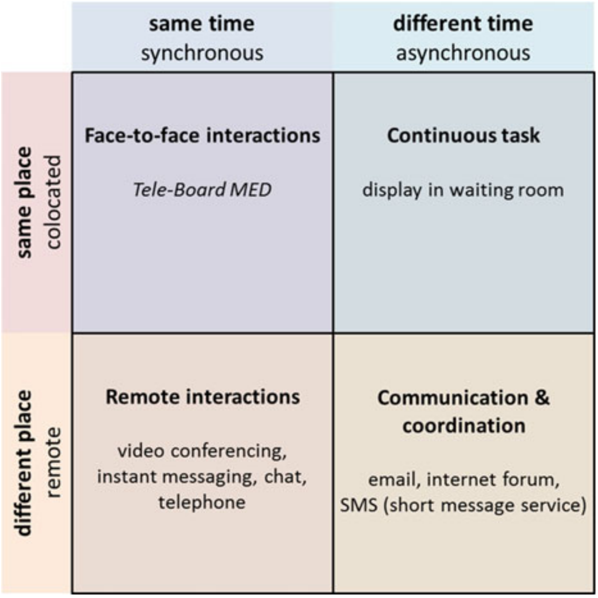


Fig. 4 The time/space matrix of Computer Supported Collaborative Work (cf. Johansen 1988) filled with groupware examples from the healthcare domain concerning collaboration between patient and provider

One way to describe the nature of cooperative eHealth applications is to locate the patient-provider interaction in space and time. The CSCW matrix by Johansen (1988) builds on these two dimensions and differentiates between four settings: same time and same place, different time and different place, same time and different place, and different time and same place (Fig. 4).

There are situations where the doctor's consultation cannot or does not have to be done face-to-face (e.g., when the patient wants to ask a question, deliver some kind of homework assignment or report, or prefers anonymous consultation). Most of the eHealth solutions that support patient-provider collaboration are designed for remote interaction. Instant messaging or videoconferencing tools can support patient and provider in synchronous interaction over distances. Via email they can participate in communication and coordination from different places and at different points in time. Looking at the upper row of the CSCW matrix dedicated to a colocated scenario, one can see that groupware used in the same place by patient and provider is relatively sparse.

Interestingly, the paper-based patient record can be seen as one of the earliest artefacts used to support patient-doctor collaboration (Fitzpatrick and Ellingsen 2012). It is mobile and can be brought into the discussions. Yet, it might be less than ideal if the provider's handwritten notes are illegible for the patient. Electronic patient records on the other hand contain digital—and thus legible—notes, but, most of the times, they are only understandable to the care provider and other health care staff. A data-centered design prevails for electronic records; data classes and categories have been identified in order to support the care provider's documentation. However, the practices in which documents are written, read, and used within the consultation have been largely ignored (Heath and Luff 1996). This is why electronic records often fail to enhance the clinical practice and its goal of patient-centeredness. Therefore—whether the patient sees unreadable handwriting or the back of a computer screen—current medical record keeping falls short in supporting the interactivity of the medical encounter.

We believe that there is unexplored potential for eHealth technologies to make the medical encounter itself more cooperative and more engaging for the patient. Therefore we propose a tool that makes a change in patient-provider interaction. This tool is Tele-Board MED.

The medical documentation system Tele-Board MED builds on Tele-Board, a web-based software system to support creative collaboration for design thinking teams (Gumienny et al. 2011; Gericke et al. 2012; Wenzel et al. 2014). With Tele-Board, the shared working area is no longer a writable, magnetic board; but instead a big, touch-sensitive screen that shows digital whiteboard panels is used. At the same time, team members stick to familiar working modes. With Tele-Board they can create sticky notes, draw scribbles, include photos and pictures, and arrange contents according to the current working step and process phase with the swipe of a finger. On this basis, Tele-Board MED is envisioned as a unique tool that supports exactly those features that turn out to be important for positive patient-provider interactions.

4 Tele-Board MED: Medical Documentation System and Research Project

The interactive documentation system Tele-Board MED is designed as an adjunct to medical encounters. It is a means for provider and patient to create a visually enhanced and freely editable patient record. Documentation can be done instantly during the encounter with the patient invited to contribute. We think that the encounter dialogue can be enhanced by involving the patient in the documentation and by providing a graphical presentation of the personal clinical picture. More specifically, we believe that Tele-Board MED addresses the factors of a high quality patient-provider relationship, as elaborated in Sect. 2, namely the patient-centered factors (Fig. 1) and the process-related factors (Fig. 2).

In our research project we also look into other, non-technical possibilities to redesign the clinical encounter. We take the design thinking mindset into healthcare and try out methods and tools from creative, user-centered team work with patients.

Mental Health Care as a First Field of Application The domain of medical encounters is huge and manifold. A research project, however, needs to start at some point. The Tele-Board MED research team decided to focus on psychotherapy as a first field of application. In this domain, patient-provider interactions are particularly well researched. Practitioners take care to regularly observe the tone of their interactions with patients. One reason is that a positive patient-provider relationship is known to be a major predictive variable for treatment success, as reflected in health improvements on behalf of the patient (Grawe 2005). In addition, one central goal in mental health care is to increase the patient's self-efficacy (Bandura 1977), which implies that psychotherapists already aim at patient empowerment in a broad sense (Aujoulat et al. 2006; Barr et al. 2015). Finally, patient empowerment seems to be particularly beneficial in the management of chronic diseases, such as diabetes and cancer. Neuropsychiatric disorders account for over 40 % of all chronic diseases and are the biggest cause of years lived with disability (World Health Organization 2005). All of these factors suggest that Tele-Board MED can be studied well in the domain of psychotherapy. For psychotherapists, innovations that support positive patient-provider interactions and patient empowerment are especially interesting. Psychotherapists are already very attentive to these issues and likely to spot any positive or negative effects of Tele-Board MED immediately. If positive effects can be achieved with the system, documenting treatment sessions with Tele-Board MED might actually make patients healthier. This would be a highly valuable and maybe surprising "side-effect" of a medical documentation tool.

User-Centered Field Research While a lot of technology is developed by engineering experts in a laboratory far away from the users, this has not been the case for Tele-Board MED. The project started in 2012 as part of the Hasso Plattner Design Thinking Research Program. Regarding both product development practices and theoretical objectives, the project has always been driven and inspired by design thinking.

One important idea of design thinking is to leave the laboratory and spend a lot of time with the people who might need and use a new tool. First, this means entering the situation as it is at present, without new tools. By observing people, talking to them, and personally doing what they do, design thinkers try to understand important needs of stakeholders and potentially the hindrances they face.

Later on it is important to leave the laboratory once again. Design thinkers turn product ideas into prototypes quickly and test them with users. One central motto is to "fail early and often" since early failures can help teams to learn rapidly and produce better fitting solutions in the end.

Accordingly, one member of the Tele-Board MED team immersed herself thoroughly in the domain of psychotherapy. To tap a somewhat broad range of psychotherapeutic encounters, she entered three kinds of scenarios: (1) a small psychotherapeutic group practice, (2) an outpatient psychotherapeutic clinic, where about 100 therapists work, and (3) a hospital ward for inpatient psychiatric treatments. In each of the three settings, she has so far spent more than 500 h learning about the institutions and administrative processes as well as empathizing with the stakeholders. Due to her own professional background, she was also allowed to see patients herself. In the later phase of the project this meant that Tele-Board MED could be used for documentation together with patients, and it could be iteratively improved on the basis of feedback. At present there have been more than 1500 h of user-encounters to learn about the needs and hindrances of patient-provider interactions in psychotherapy—when Tele-Board MED is not used and when it is used.

Design Thinking Process In general, the design thinking work methodology builds on a loosely defined iterative problem-solving process. In the following we will report some key findings of the Tele-Board MED project using a revised process model (von Thienen and Meinel 2014) with five steps: (1) Empathize, (2) Define Point of View, (3) Ideate, (4) Test Prototypes, (5) Bring Home.

4.1 Step One: Empathize

In the phase of empathizing, design thinkers immerse themselves thoroughly in the domain that they want to design for and that they want to understand. In the case of this project, we have spent a lot of time with psychotherapists and therapy patients. These are some key insights from our encounters:

Starting with Psychotherapists we know that their primary concern is to help patients. However, in all three scenarios (small group practice, outpatient and inpatient clinic), therapists do not spend their full work day seeing and helping patients. Rather, a striking amount of time is spent on administrative tasks. In outpatient treatments, a very time-consuming task involves writing medical reports. This means describing the patient's case, analyzing the patient's problems and suggesting a treatment plan. Such a medical report has to be sent regularly to the patient's insurance company to get funding. In the case of inpatient treatments, therapists write discharge letters of comparable length. The task of writing such reports and letters is all the more annoying for therapists since they often end up writing down the same information more than once. During therapy sessions they take handwritten notes. To obtain official reports, they have to type all the information into a computer. (For more information regarding the therapists' view on administrative duties and on their documentation habits, please see von Thienen et al. 2015; Perlich and Meinel 2015.)

Patients on the Other Hand typically experience great strain in therapy sessions. They long for an understanding and supportive professional with whom they feel safe enough to confront overwhelming life problems. However, to tackle personal problems, patients need to reveal private or even "secret" thoughts and feelings. The therapist may nod smilingly, yet the patient does not see what exactly he writes down when taking notes. It could well be an interpretation that the does not satisfy the patient. The therapist may note that the patient "shows dysfunctional behavior" or "narcissistically demands approval." Very often patients learn about the interpretations of their therapists only after a number of sessions, for instance, when hearing a diagnosis or reading their own hospital discharge letter. Furthermore, many psychotherapy patients report "bad experiences" with past therapeutic care providers. In earlier encounters, they felt misunderstood, or generally "not in good hands." They are accordingly very skeptical towards a new therapist. Not knowing what the therapist writes down in the patient's medical file delays the process of trust building.

4.2 Step Two: Define a Point of View

When defining a point of view, design thinkers mold their experiences with many stakeholders into single stories of concrete, albeit partly fictitious "personas". For Tele-Board MED, let us consider the following story that introduces two personas *Linda* and *Dr. Bernstein*.

Linda is a 25-year old linguistics student who suffers from anxiety attacks. At some point in her life, she feels she needs psychotherapeutic support. Linda rarely finds the courage to leave her apartment any more. She is too afraid of having panic attacks in public. Her boyfriend even has to do the grocery shopping for her.

Linda arranges meetings with behavior therapist Dr. Bernstein (Fig. 5). He asks Linda a lot of questions about her problem and also about her biographical background. He often takes notes with his pen on sheets of paper that he keeps on his lap. Linda cannot read Dr. Bernstein's notes and wonders what he writes down. Maybe he writes things like "hysterical," "craving for attention," or something else she would not like. Linda is asked about the dates of major life events, but she is unsure about some details. Nonetheless, Dr. Bernstein notes all the dates she tells him. Linda is unsure how correct the biography is that he has written down in his file this way.

From Dr. Bernstein's perspective, there are two priorities. On the one hand, he needs to explore Linda's case and produce an adequate diagnosis and treatment plan. Since he sees a lot of patients and cannot recall every detail, he has to take notes. On the other hand, Dr. Bernstein wants to establish a relationship of trust so that Linda feels comfortable revealing delicate issues of her life. Dr. Bernstein would like to pay full attention to Linda throughout the sessions. He feels uncomfortable having to interrupt her at times so he is able to take notes. He tries to write very fast. However, later on he is sometimes unable to read his own scribbles.

Over time, Linda and Dr. Bernstein work intensively on Linda's problems. Together, they understand why Linda behaves and feels the way she does. In addition, they develop new ideas about how Linda could confront her problem situations to overcome anxiety issues. Linda is quite satisfied with these meetings. However, she feels there are many important things they discuss in their sessions that she would like to recall later on. The entire treatment

(continued)

Fig. 5 Linda sees therapist Dr. Bernstein who documents the sessions with handwritten notes on sheets of paper

Fig. 6 In the days after her therapeutic encounter, Linda tries to recall her last therapy session. However, no session documentation is available to her. Thus, she does not recall all the important things that were said. Her memories remain for the most part nebulous

documentation, however, remains in Dr. Bernstein's practice. She is uncertain about what exactly they said, whether she can recall everything correctly, or if she might forget something crucial. Thus, therapy sessions seem somewhat elusive to Linda. Looking back on her sessions, Linda's thoughts sometimes seem caught in what might be described as the whirl of a carousel or they simply remained hidden in a fog (Fig. 6).

Dr. Bernstein is a passionate therapist. He would like to spend all of his workday seeing patients. However, to obtain funding for treatments, he has to write lengthy case reports. Since he is an experienced therapist, he is

(continued)

Fig. 7 Dr. Bernstein goes through piles of handwritten notes to create a case report. What he wrote down by hand in the past, he now types into a computer

obligated to write the first report after 30 treatment sessions (as a novice therapist, he would already have to produce a report after the fifth session). Linda and Dr. Bernstein have been meeting once a week. Therefore, Dr. Bernstein has to now sift through notes from more than half a year of treatment. There are piles of pages on his table. Unfortunately, there is no automatic search function for his handwritten documentation. He has to go through his scribbles, trying to understand things that he wrote down a long time ago. He then has to write down the same information once again, this time in a digital format, to obtain an official case report (Fig. 7).

4.3 Step Three: Ideate

In the phase of ideation, design thinkers explore many different ideas of how to help the stakeholders satisfy their fundamental needs and overcome obstacles. In the case of our project, the design thinking tool Tele-Board was chosen as a basic platform for new suggestions. Many very diverse ideas were developed on how to support patient-therapist interactions with Tele-Board MED, based on existing or new functionality. These are some key ideas and visions:

Fig. 8 Linda and Dr. Bernstein compile and use case information jointly

Fig. 9 Linda takes a printout of the medical documentation home. Thus, she can check the correctness and completeness of all details. Furthermore, she can go back to the most important insights of the sessions. This means she won't forget them

Using Tele-Board MED, the patient file is visible to both Dr. Bernstein and Linda. Linda can see the keywords her doctor writes down. When he misunderstands something she can intervene immediately. Notes and scribbles can help the two of them to exchange knowledge and collaborate (Fig. 8).
However, the benefits of Tele-Board MED should not be limited to those moments when therapist and patient interact directly.
After each session Dr. Bernstein prints out a copy of the documentation panel with the latest notes on a sheet of paper, which Linda can take home. Thus, she can check whether all the information is correct, which she reported in the

(continued)

Fig. 10 Dr. Bernstein transfers the medical case information with one click from Tele-Board MED to an MS Word file

anamnesis session. In addition, she can think about whether or not some important details might be missing. Finally, she can read what exactly was said about her problem and how she could approach things differently (Fig. 9). When writing a case report, Dr. Bernstein can use the Tele-Board MED export function. Immediately, all the relevant case information is transferred to an MS Word file containing text and pictures. He just needs to turn the key words into a running text (Fig. 10).

4.4 Step Four: Test Prototypes

In testing phase, design thinkers bring one or more prototypes out in the field, observe users, and invite them to try out the prototype and comment on it.

In the case of our project, there have been two types of "reality checks." First, there have been tests of Tele-Board MED features based on paper prototypes and different versions of software and hardware. Second, we have tried out different ways of introducing Tele-Board MED to therapists, who we have then trained in system usage.

While some features suggested for Tele-Board MED are considered more helpful than others (von Thienen et al. 2015), the general concept is highly appreciated. Therapists state that they can save one third of their normal working time, or even more, when writing case reports. Furthermore, the system is considered the best tool available at present to meet the new legal requirements of patient empowerment. In Germany a federal law on patient rights (Bundesgesetz 2013) requires that medical records be accessible and obtainable for patients. This seems an almost impossible task based on the traditional, handwritten case documentation, which the patient might not even be able to decipher if given the opportunity.

To test real software and hardware with patients in therapy sessions, a lot of technical preparation was necessary. A fundamental prerequisite in medical documentation is to store patient data in a secure way. Therefore, we added additional security features to the platform and setup of Tele-Board MED to keep the data confidential and protect it from being compromised (Perlich et al. 2015).

Patients' Feedback on Tele-Board MED The patients' feedback was very positive. They were very happy to see their doctor's notes. This transparency also seemed to contribute to the understanding of and concordance with therapeutic treatments. For instance, in the hospital context, patients typically come with severe illnesses and a long record of (moderately successful) treatments in the past. One patient said: "I have had this diagnosis for more than 10 years. Thank you so much. For the first time I understand why I get such a diagnosis." (von Thienen, personal case documentation).

Furthermore, several patients were treated with Tele-Board MED who had obtained an unpopular diagnosis in the past. Indeed, these patients had rejected their diagnosis so resolutely that their former doctors refrained from handing out their letters of discharge—which revealed the diagnosis and a corresponding treatment. Given that doctors have a legal obligation to issue discharge letters, with rare and legally complicated exceptions, this is a striking intervention on behalf of the care providers and suggests a high degree of patient-doctor conflict. In all cases, the patients now treated with Tele-Board MED obtained the same diagnosis once again. However, as the diagnostic procedure was made transparent to them, every patient agreed on the diagnosis in the end and embraced corresponding treatments.

In general, we have consistently seen positive patient-provider relationships when Tele-Board MED was used. There have been no cases of open patient-therapist conflict. Furthermore, typical indicators of underlying conflicts (such as an unheralded absence of the patient in a scheduled session or a complete therapy drop-out) have not been observed in treatments with Tele-Board MED so far. Quite to the contrary, patients show considerable and uncommon teamwork behavior. Almost every patient helps to arrange the room (adjust the light, close the door, help carry therapy equipment etc.) after three or less sessions with Tele-Board MED. Furthermore, the German language has a built-in "relationship detector." There are two ways of addressing other people. The formal and official form of "you" is "Sie". This is the way doctors and patients normally address each other. However, friends, family members or close acquaintances use the familiar form of "you"—"du". In treatments with Tele-Board MED, every second patient accidentally addressed the doctor with "du" at least once and immediately excused him or herself afterwards for the slip up. We take this as a very strong indicator of a "team feeling" since it is quite an uncommon observation in sessions without Tele-

Board MED. In terms of patient empowerment, we noted that every third patient spontaneously stated that he or she would like to take home a complete copy of the personal medical record at the end of the treatment. This is also very uncommon.

Therapists' Feedback on Different System Setups Apart from using Tele-Board MED with patients, something quite different has been prototyped and tested: the learning experiences of therapists. How can they learn the proper handling of Tele-Board MED?

First of all, there is not one single system setup. Tele-Board MED is a software application that can be used on a lot of different hardware devices. The flexibility to adapt to different situations has been taken over from the mother system Tele-Board (Gumienny et al. 2012). We have set up and iterated two different setups. One is rather comprehensive, including a large, digital, touch-sensitive whiteboard, a mouse, digital pens, two keyboards and two tablet computers (Fig. 11). Another setup is comparably basic, consisting of a laptop, a projector and one keyboard with touchpad (Fig. 12).

In terms of trainings, we started with 3-h workshops in each of which a small group of approximately five therapists was introduced to the system and could try it out in small exercises. However, this kind of training did not lead to therapists using the system with their patients—for understandable reasons. First, therapists want to pay full attention to their patients (and not to documentation tools). After all, a good rapport with the patient is essential for treatment success; and therapeutic mistakes can have devastating effects, such as the suicide of a depressive patient. Furthermore, therapists face extensive bureaucratic obligations including documentation duties that leave little room for trying something out that might potentially fail. In addition, patients can be very sensitive to the slightest signs of therapeutic incompetence, and therapists do not wish to come across as novices who don't understand their work. Unfamiliar technology, such as a digital whiteboard or even a tablet computer, always harbors the risk that something might not work the way it is supposed to. Finally, even when the program works correctly, its operation needs to be learned. Therapists do not want to stand in front of their patients wondering what button to press while time goes by and the patient could receive a sufficient treatment without digital support.

Thus, fear of unforeseeable technical problems is a great barrier. A 3-h workshop does not provide enough training time for therapists to feel comfortable using the system—even though they say they could operate the software intuitively. A second issue is time shortage. Both in the outpatient and inpatient clinic, rooms are booked for short timeslots (either 30 or 60 min). The therapists don't have time for a 5-min system setup (or even longer); they need to start their work with the patient immediately. Therefore, both the hardware setup and therapist trainings are currently iterated to yield more viable versions.

Fig. 11 A "comprehensive" Tele-Board MED setup that includes a digital whiteboard, tablet computers, and mobile keyboards

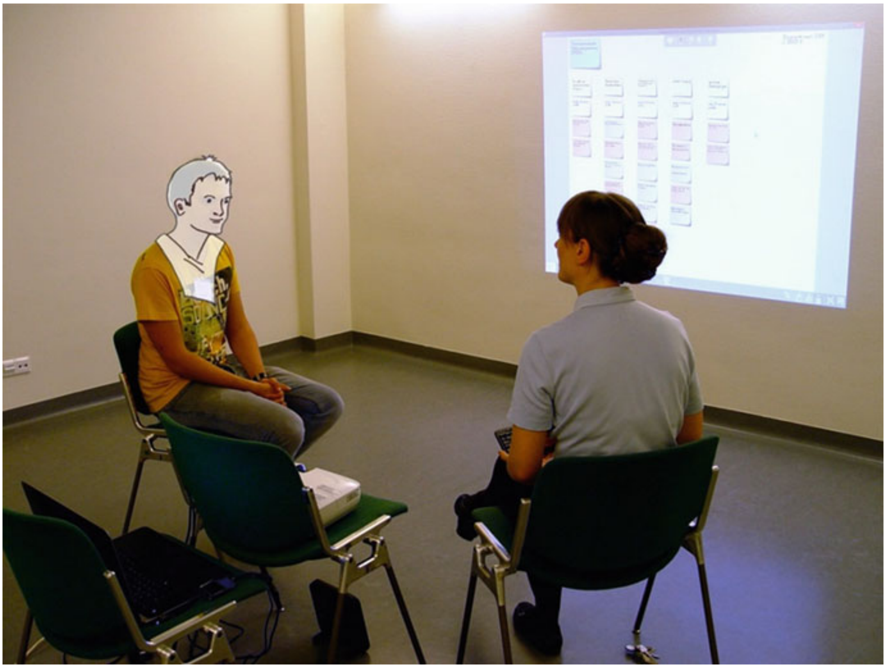


Fig. 12 A "light" Tele-Board MED setup using laptop and projector

4.5 Process Step Five: Bring Home

In the phase of bringing an idea home, a prototype is turned into a product. That means it becomes officially accessible for a broad range of stakeholders.

The Tele-Board MED project is only at an early stage of bringing the idea home. The creation of a Tele-Board MED prototype that is thoroughly useful and has good usability characteristics in all three scenarios (small practice, outpatient and inpatient clinic) is already well advanced. However, the development of practical training modalities for therapists is still in progress. The question of how Tele-Board MED can become reality in a broad range of medical encounters is the next to follow.

5 Summary

To conclude our research, we will summarize how Tele-Board MED as an eHealth tool for face-to-face medical encounters can contribute to patient empowerment. Building on our suggested unified patient empowerment model (Fig. 3), we see how the patient-centered as well as the process-centered aspects of patient-provider relationships are met.

Patient Knowledge The Tele-Board MED documentation panels are visually enhanced and serve as a medium for patient education about medical knowledge as well as a medium to collect and reflect case-specific data.

Patient Autonomy The copy or print-out of the session notes that can be taken home increases the patient autonomy. In possession of written material, the patient can follow up on the treatment between medical encounters. Furthermore the switch to another care provider can be supported as patients can take their "history" with them. The overview of the clinical picture supports an informed participation in decision-making.

Patient Values The confrontation of patients with a visual and textual presentation of their health situation, leads to a stronger verbal and intellectual debate concerning personal beliefs in the context of their care. Furthermore, we have developed several documentation panel templates for psychotherapy sessions (such as anamnesis templates) where patients are invited to explicate values (e.g., personal therapy goals).

Collaboration Process The joint documentation process calls for collaboration. Especially in psychotherapy the contributions to the encounter by patient and therapist are equally important. The graphical user interface of Tele-Board MED is based on real world gestures of teams working on whiteboards, and thus explicitly dedicated to collaboration.

Integration Process The hands-on, shared working space stimulates documentation in everyday speech, capturing via key words and visual clues. Medical jargon seems out of place here. Print-outs of session notes can be taken home and used to prepare for the next encounter. Therefore, the patient is put in a more active role from the start.

Communication Process The visibility of case notes builds a common ground for discussions. Both patient and provider negotiate on what to capture and in which way. In turn, the communication also ensures correctness and completeness of the notes. The display of information, pictures and templates can also spark the communication or take it in certain directions.

Empowerment Process All the above factors contribute to a higher empowerment of the patient. The patients are supported in building knowledge about their health situation; and once communication and collaboration happens at eye-level, patients are empowered to make informed health decisions and transform their knowledge into action.

References

Agarwal AK, Murinson BB (2012) New dimensions in patient-physician interaction: values, autonomy, and medical information in the patient-centered clinical encounter. Rambam Maimonides Med J 3(3):1–11

Aujoulat I, D'Hoore W, Deccache A (2006) Patient empowerment in theory and practice: polysemy or cacophony? J Patient Educ Couns 66(1):13–20

Bandura A (1977) Self-efficacy: toward a unifying theory of behavioral change. Psychol Rev 84 (2):191–215

Barofsky I (1978) Compliance, adherence and the therapeutic alliance: steps in the development of self-care. Soc Sci Med 12:369–376

Barr PJ, Scholl I, Bravo P, Faber MJ, Elwyn G, McAllister M (2015) Assessment of patient empowerment: a systematic review of measures. PLoS One 10(5):1–24

Bundesgesetz (2013) Gesetz zur Verbesserung der Rechte von Patientinnen und Patienten, Bonn

Emanuel EJ, Emanuel LL (1992) Four models of the physician-patient relationship. J Am Med Assoc 267(16):1–13

Fitzpatrick G, Ellingsen G (2012) A review of 25 years of CSCW research in healthcare: contributions, challenges and future agendas. Comput Supported Coop Work 22:609–665

Gericke L, Gumienny R, Meinel C (2012) Tele-board: follow the traces of your design process history. In: Platter H, Meinel C, Leifer L (eds) Design thinking research: studying co-creation in practice. Springer, Berlin, pp 15–29

Grawe K (2005) Empirisch validierte Wirkfaktoren statt Therapiemethoden. Report Psychologie, 7(8), 311

Gumienny R, Gericke L, Quasthoff M, Willems C, Meinel C (2011) Tele-board: enabling efficient collaboration in digital design spaces. In: Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp 47–54

Gumienny R, Gericke L, Wenzel M, Meinel C (2012) Tele-board in use: applying a digital whiteboard system in different situations and setups. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research: measuring performance in context. Springer, Berlin, pp 109–125

Heath C, Luff P (1996) Documents and professional practice: “bad” organisational reasons for “good” clinical records. In: Proceedings of the Conference on Computer Supported Cooperative Work, pp 354–363
HIMSS (2003) HIMSS e-Health SIG white paper. White Papers. Retrieved in January 2016 from http://www.longwoods.com/content/20034
Johansen R (1988) Groupware: computer support for business teams. The Free Press, New York, NY
Kim SC, Boren D, Solem SL (2001) The Kim alliance scale: development and preliminary testing. Clin Nurs Res 10(3):314–331
Kim SC, Kim S, Boren D (2008) The quality of therapeutic alliance between patient and provide predicts general satisfaction. Mil Med 173(1):85–90
Koch S (2012) Improving quality of life through eHealth: the patient perspective. Stud Health Technol Inform 180:25–9
Murray E, Lo B, Pollack L, Donelan K, Catania J, Lee K, Zapert K, Turner R (2003a) The impact of health information on the Internet on health care and the physician-patient relationship: national U.S. survey among 1.050 U.S. physicians. J Med Internet Res 5(3), e17
Murray E, Lo B, Pollack L, Donelan K, Catania J, White M, Zapert K, Turner R (2003b) The impact of health information on the internet on the physician-patient relationship: patient perceptions. Arch Intern Med 163(14):1727–1734
Perlich A, Meinel C (2015) Automatic treatment session summaries in psychotherapy: a step towards therapist-patient cooperation. Procedia Comput Sci 63:276–283
Perlich A, von Thienen J, Meinel C (2014) Supporting talk-based mental health interventions with digital whiteboards. Stud Health Technol Inform 205:433–437
Perlich A, Sapegin A, Meinel C (2015) Implementation of data security requirements in a web-based application for interactive medical documentation. In: Proceedings of the International Conference on Health Informatics (HealthInf 2015), pp 352–359
Schmidt K, Bannon L (1992) Taking CSCW seriously: supporting articulation work, Cooperative work and coordinative practices. Springer, London, pp 45–71
von Thienen JPA, Meinel C (2014) Let’s bring home even more benefits from design thinking: ideas for an iterated design thinking process model. Electronic colloquium on design thinking research, http://ecdtr.hpi-web.de/report/2014/001
von Thienen JPA, Perlich A, Meinel C (2015) Tele-board MED: supporting twenty-first century medicine for mutual benefit. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research: building innovators. Springer, Berlin, pp 101–130
von Thienen JPA, Perlich A, Eschrig J, Meinel C (2016) Smart documentation with tele-board MED. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research: making design thinking foundational. Springer, Berlin, pp 203–233
Wenzel M, Gericke L, Gumienny R, Meinel C (2014) Creative collaboration in real world settings. In: Plattner H, Meinel C, Leifer L (eds) Design thinking research: building innovation eco-systems. Springer, Berlin, pp 115–134
Wilson P (1991) Computer supported cooperative work: an introduction. Intellect Books, Oxford, UK
World Health Organization WHO (1986) The Ottawa charter for health promotion. Retrieved in January 2016 from http://www.who.int/healthpromotion/conferences/previous/ottawa/en/
World Health Organization WHO (2005) Mental health: facing the challenges, building solutions, Report from the WHO European Ministerial Conference

Embodied Design Improvisation for Autonomous Vehicles

David Sirkin, Sonia Baltodano, Brian Mok, Dirk Rothenbücher, Nikhil Gowda, Jamy Li, Nikolas Martelaro, David Miller, Srinath Sibi, and Wendy Ju

Abstract We have developed a generative, improvisational and experimental approach to the design of expressive everyday objects, such as mechanical ottomans, emotive dresser drawers and roving trash barrels. We have found that the *embodied design improvisation* methodology—which includes storyboarding, improvisation, video prototyping, Wizard-of-Oz lab studies and field experiments—has also been effective in designing the behaviors and interfaces of another kind of robot: the autonomous vehicle. This chapter describes our application of this design approach in developing and deploying three studies of autonomous vehicle interfaces and behaviors. The first, *WoZ*, focuses on the conceptual phase of the design process, using a talk-aloud protocol, improvisation with experts, and rapid prototyping to develop an interface that drivers can trust and hold in esteem. The second, the *Real Road Autonomous Driving Simulator*, explores people's naturalistic reactions to prototypes, through an autonomous driving interface that communicates impending action through haptic precues. The third, *Ghost Driver*, follows the public deployment of a prototype built upon frugal materials and stagecraft, in a field study of how pedestrians negotiate intersections with autonomous vehicles where no driver is visible. Each study suggests design principles to guide further development.

1 Introduction

> Car: "*So, I'm noticing that one of my tires is low on air. Is there something about yourself that you want to talk about?*" Driver, after two seconds, laughs out loud: "*That's the most bizarre question I've ever heard from a car.*"

D. Sirkin (✉) • S. Baltodano • B. Mok • D. Rothenbücher • N. Gowda • J. Li • N. Martelaro • D. Miller • S. Sibi • W. Ju
Center for Design Research, Stanford University, 424 Panama Mall, Stanford, CA 94305, USA
e-mail: sirkin@stanford.edu

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_9

Driver: "*Tell me about yourself, car*." Car: "*What would you like to know?*" Driver: "*Where are you from?*" Car: "*I'm from Japan. Where are you from?*" Driver, after five seconds: "*This is like a* 'Her' *moment*."

-Conversations Between Drivers and Autonomous System WoZ Study

We have previously written about our method of *embodied design improvisation* to design machines and robots where (a) physical motions, gestures or patterns are employed, (b) the design space of possible actions, mechanisms and dimensions is vast, and (c) the cost in time, money and effort to build fully functioning systems is high (Sirkin and Ju 2015). This method exemplifies the challenges of design thinking, where the understandings that designers draw upon are often tacit, where an exhaustive search of the design space is not possible, and where the costs of full-scale solutions precludes iterative empirical testing.

The context for these prior articles was the form and movement of *expressive everyday objects*: non-anthropomorphic household furnishings that can initiate, conduct, and conclude interactions with people in a meaningful, improvisational way. In this chapter, we illustrate how we have applied the same approach to the design of interactions in another domain: autonomous vehicle interfaces.

Each of the following three sections summarizes a behavioral experiment in this new context, and focuses on aspects of the *embodied design improvisation* process: *WoZ* relates to early-stage exploration of design ideas within the vehicle through improvisation and enacted scenarios, the *Real Road Autonomous Driving Simulator* highlights rapid prototyping of physical interfaces applied to a field experiment, and *Ghost Driver* underscores the iterative redesign of study procedures to observe and understand pedestrian interactions outside of the vehicle.

2 Case Studies in Design Improvisation

These case studies center on the design of interactions between people and *near future* autonomous vehicles: in particular, how vehicles that operate with varying degrees of autonomy—from basic driver assistance to fully self-driving systems—can, and should, interact with drivers and passengers inside of the vehicle, and with pedestrians and bicyclists outside of it. Over the course of these projects, our goals have been (a) to observe and understand how people respond to vehicles that exhibit intrinsic agency during everyday driving, and (b) to develop and explore vehicle interfaces and behaviors that express that agency, including their features and limitations.

Each study draws upon *Wizard of Oz* techniques, which are frequently employed in interaction design, and which we further motivate and describe in the next section. Experimenters operate as *stand-ins* for future technologies that would otherwise perform extended, contingent interactions with people. The name comes from the novels of L. Frank Baum, wherein a Wizard is believed by all of the denizens of the Land of Oz to be a magical being, where in fact, he is an ordinary man employing a variety of tricks to project an illusory reality (Baum 1900).

2.1 *Wizard of Oz (WoZ)*

2.1.1 Introduction

The WoZ system (Mok et al. 2015) is designed (a) to explore how drivers and their partially or fully autonomous vehicles can share and exchange driving tasks, and (b) to understand the thoughts and feelings that drivers experience during these transitions. Driving tasks include controlling the vehicle as it rolls down the road, or navigating through town to some destination. A transition is where the driver indicates that the car should assume some responsibility that he or she currently holds and the car accepts that role, or the reverse: where the car indicates and the driver accepts. Taking this perspective, driver and car act as a team (Inagaki 2009), collaborating to reach their destination in a safe, legal, timely and comfortable way. It thus becomes important that they learn each other's abilities, and communicate and understand each other's intentions and actions. By integrating WoZ into a driving simulator (see Fig. 1), we can alter the car's abilities and actions at a moment's notice, or update the surroundings (including pedestrians, other cars, and their behaviors) from one drive to the next.

2.1.2 Prototype Systems

The WoZ station allows researchers to communicate with participant drivers in the simulator vehicle through a speech interface, as well as to initiate or respond to transfers of control and operate the vehicle during periods of autonomous driving (see Fig. 2). In fact, the vehicle does not drive autonomously: rather, researchers control the car's actions using a Wizard of Oz protocol (Dahlbäck et al. 1993; Cross 1977), where drivers are told that they are interacting with an autonomous system, but their interactions are mediated by a human operator. As noted by Hoffman and Ju (2014), this approach allows us to explore a wide range of features and functions without first building a fully operational system.

Due to the challenge of attending to several simultaneous tasks, WoZ has dual control stations. The Interaction Wizard observes and interacts with the participant using video cameras and a text-to-speech interface, making the car appear to be able to detect, and respond, to the drivers' movements, facial expressions and utterances. The Driving Wizard controls the car's autonomous driving, triggers events in the simulated environment—such as pedestrians crossing the road or surrounding cars braking quickly, and updates elements of the car's visual interface—such as instruments panels.

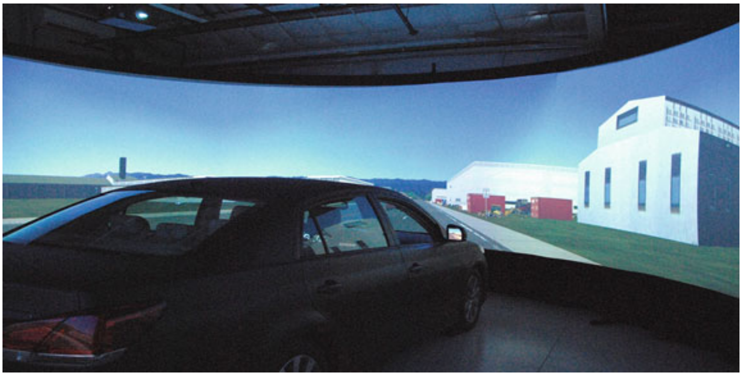


Fig. 1 The simulator used for experiments in autonomous vehicle driving and control. Participants sit in a fixed-base car surrounded by a 270° screen depicting the study environment

Fig. 2 The Wizard of Oz control station. Wizard 1 (the Interaction Wizard) interacts with the participant, while Wizard 2 (the Driving Wizard) initiates transfers of control and operates the vehicle during autonomous driving mode

2.1.3 Designed Behaviors

From the driver's perspective, the car presents as a physical, social agent, able to (a) operate autonomously, with varying degrees of competence (which, unbeknownst to the driver, may be high or low, depending on the researchers' agenda), (b) carry on spoken dialog, in a male or female voice, about driving topics such as the roadway and navigation, or notably, non-driving topics such as the driver's preferences, expectations, experiences or emotional state, and by combining these (c) initiate and respond to (for example) requests to change speed if running late, detour for lunch, or transfer control if the driver feels sleepy.

2.1.4 Improvisation Sessions

We invited 12 interaction and interface design experts to act as participants in individual design improvisation sessions of about 30 min each. Given the exploratory nature of the study, we provided participants with little information about the car or instruction for interacting with it, and encouraged them to actively improvise to discover its abilities.

Participants traveled through a course comprising four sections: (a) a brief practice to introduce the simulator and its environment, which includes straight roads, intersections and roundabouts, (b) a stretch of forests and hills, (c) a city with densely placed buildings, pedestrian intersections and crossing traffic, and (d) a highway. Notably, in each of the latter three sections, several potentially dangerous events occurred: car cutoffs, pedestrian incursions and cars pulling onto the road. During the drive, including these events, the car offered to assume control at certain times, and requested that participants resume control at other times. The car (at the behest of its Wizards) did not follow a controlled response protocol: rather, it freely offered explanations for its driving (*pushing* information to drivers), or responding to queries about its behavior (*pulling* information from drivers). We recorded the entire exchange, and interviewed participants after the session about their preferences, experiences and thoughts about the events and car's responses.

Desire for Shared Control

On several occasions, the car intentionally drove *imperfectly*, drifting laterally within its lane, crossing into the sidewalk during a turn, or closely approaching people or cars directly ahead. Although the car's performance was flawed, for the most part, participants still held the system in esteem, and preferred to make gentle corrections to the pedals or steering wheel, and remain in autonomous mode.

This form of shared control is not considered under National Highway Traffic Safety Administration's current Levels of Automation model (National Highway Traffic Safety Administration 2013). The design challenge therefore becomes how shared control *should* function. Participants expected that the car would modify its behavior based on their guidance. But how can the car know when that guidance had ended? Also, some participants kept their hands on the steering wheel as a way to monitor the car's actions, or to lessen their feelings of unease, rather than as a way to prime the car's future behavior, making it difficult to interpret their intent. One approach may be to monitor speech, which for the current study, included comments such as "*you're drifting to the right*" or "*you're too close to the car ahead*." Such explicit signals may be unstructured, but at least they carry clear intent and can be interpreted.

Trust in Autonomy

Participants reported that two behaviors significantly improved their trust in the car's autonomous system: (a) successfully traversing challenging sections of road, including a traffic circle and an s-curve, and (b) calling out features or events in the environment *that a human would have found noteworthy*. Regarding the latter, drivers felt that noting every possible event in the environment would become tiresome, but that highlighting events related to safety, or which the driver might have wanted to observe but had missed, made the car seem more like a peer. At one intersection, the car commented that one pedestrian in a group at the crosswalk was lagging behind the others, and almost all participants interpreted that observation as the car perceiving the world the same way that they (as humans) do.

On the other hand, after experiencing more egregious bouts of imperfect driving—such as maintaining uncomfortably short headway distance—participants tended to disengage automation and disapprove requests (by the car) to resume autonomous driving. Over time, their trust in the system could be repaired, but only after ongoing, non-driving-related conversation, such as the quotes at the start of the chapter (Sirkin et al. 2016), or several small trust-building tests, such as providing a warning prior to a car cutoff. For example, after the car announced "*There are people here,*" the driver responded "*Do you know how many people, car?*"

Driving Mode Transitions

Participants often felt that the timing of transitions was unclear. In particular, even short phrases such as "*I have control now,*" when spoken by the car, are expressed over time. The resulting uncertainty over when the transition occurred, or whether it was safe to relinquish control, led participants to ask "*Are you driving the car now?*" or "*Can I let go of the steering wheel now?*" We found that adding a chime, with sharp attack, provided a better transition demarcation (Fig. 3). While participants felt that a "*3-2-1*" spoken countdown prior to the chime provided even greater advance notice, it also extended the transition time, and became a nuisance after several times.

We also tested visual indicators of mode changes, including an instrument cluster graphic which changed color from gray to green, and wording from "*Autonomy Off*" to "*Autonomy On,*" during transitions to autonomous control. Participants found such cues helpful in determining when transitions had occurred, and suggested that a haptic indicator—such as a vibrating steering wheel, or tightening seatbelt, or moving seat—might provide even more effective notification.

Addressing Requests

Participants often instructed the car to perform certain tasks, such as "*pass that slow vehicle in front of us*" or "*tell me about today's news headlines*" or "*play music on*

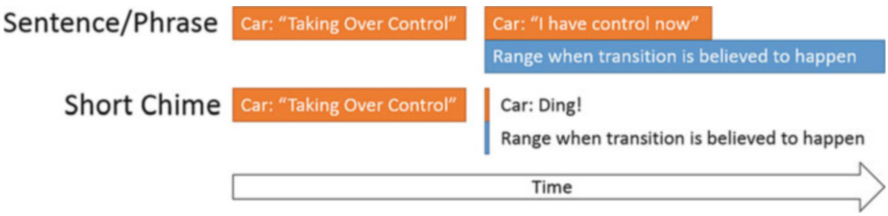


Fig. 3 At the *top*, a spoken phrase like "*I have control now*" takes time, making the moment of transition unclear. At the *bottom*, a chime communicates the exact moment more clearly

my play list," some of which might be inadvisable, and others unavailable. Drivers were typically satisfied if the car provided a technical rationale for not performing the task, such as not having access to certain audio files, but they were less willing to relent when they knew that the car was capable of performing the request, but refused to do so for some non-technical reason, such as exceeding the speed limit. In this case, participants continued to ask the car to speed up, with several eventually choosing to disengage autonomy and drive over the speed limit.

Response Latency

The car typically responded to participants' requests, commands and conversation within about 10 s, limited by Wizard 1's ability to type the message quickly. This delay was particularly noticeable for longer answers, which created an extended, uncomfortable silence, causing participants to question whether the car had heard, or interpreted, what they said. One way to ameliorate the problem is to provide a short acknowledgment, allowing for a detailed follow-up. For example, saying "*let me find out*" signals that the car has received and understood the message, and is working on a response. An alternative, suggested by one participant, is to play audio tones (such as soft beeps) that suggest that the computer is processing the information.

2.2 Real Road Autonomous Driving Simulator (RRADS)

2.2.1 Introduction

RRADS (Baltodano et al. 2015) is an on-road vehicle platform and set of study procedures to help researchers design and test autonomous vehicle interfaces. We developed the system specifically to explore attitudes and concerns that people may have in real-world, rather than fixed-base simulator, autonomous vehicles.

There are currently few platforms available to support such research. Virtual lab-based simulations excel at creating highly structured and controlled events (Talone et al. 2013), however, they are difficult to acquire and maintain, and struggle to replicate the rich sensory stimuli, inertial forces, changing lighting and weather, and unpredictable traffic patterns experienced in on-road settings. For these reasons, we were motivated to developing a low-cost, safe and reliable real-world testbed for human-autonomous vehicle interactions.

2.2.2 Prototype Systems

RRADS involves two Wizards (Kelley 1983; Dahlbäck et al. 1993) in a single vehicle: a *Driving Wizard*, who controls the vehicle from the usual driving position, and an *Interaction Wizard*, who operates the interfaces being developed from the rear seat. Three GoPro cameras record road events, participants' reactions, and the actions of the Wizards. A partition made of stiff, opaque material obfuscates participants' view of the Driving Wizard (Fig. 4).

The partition plays a dual role during experiments: preventing the participant from seeing the Driving Wizard, while not compromising the Wizard's ability to use driving controls and safety features, including steering wheel, shift lever, pedals and mirrors. It is constructed of stiff, 2 cm thick foam core board, affixed to the vehicle interior using gaffer's tape. Figure 5a, b show the partition installed in an Infiniti M45, one of our two test vehicles (the other being a Jeep Compass).

Figure 5b also shows a (non-functional) steering wheel mounted to the dashboard directly ahead of the participant. We found that even this small gesture suggests the participant's role as being more than just passenger. It also supports the participant's suspension of disbelief, and in turn, increases the effectiveness of the simulation.

Three GoPro cameras, oriented as shown in Fig. 4, record events during the experiment, with the camera shown in Fig. 5b focusing on the participant's hand motions and facial expressions. Through these, the Interaction Wizard observes the participant's reactions, allowing in-the-moment, improvisational responses.

2.2.3 Designed Behaviors

The RRADS protocol has three main sections, each of which is designed to support participants' suspension of disbelief.

Meet and Greet

At the start of a session, a researcher greets and guides the participant to the vehicle, approaching it from the passenger side (Fig. 6). The vehicle is parked along the curbside with the Driving Wizard inside, but not visible through the windows, and the Interaction Wizard waiting by the rear passenger door. The Interaction Wizard is introduced as monitoring the autonomous system, and the participant is seated.

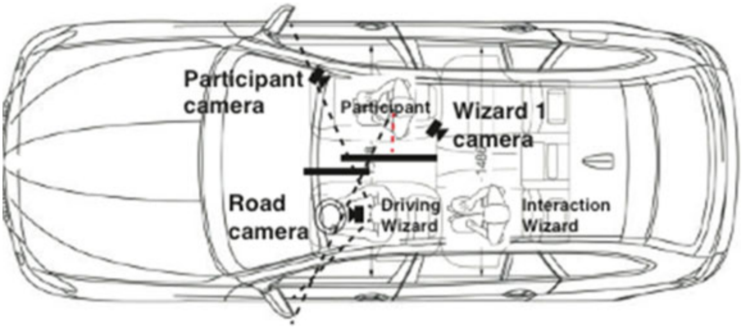


Fig. 4 The RRADS platform vehicle, noting placement of the Driving and Interaction Wizards, study participant, recording cameras, and opaque partition

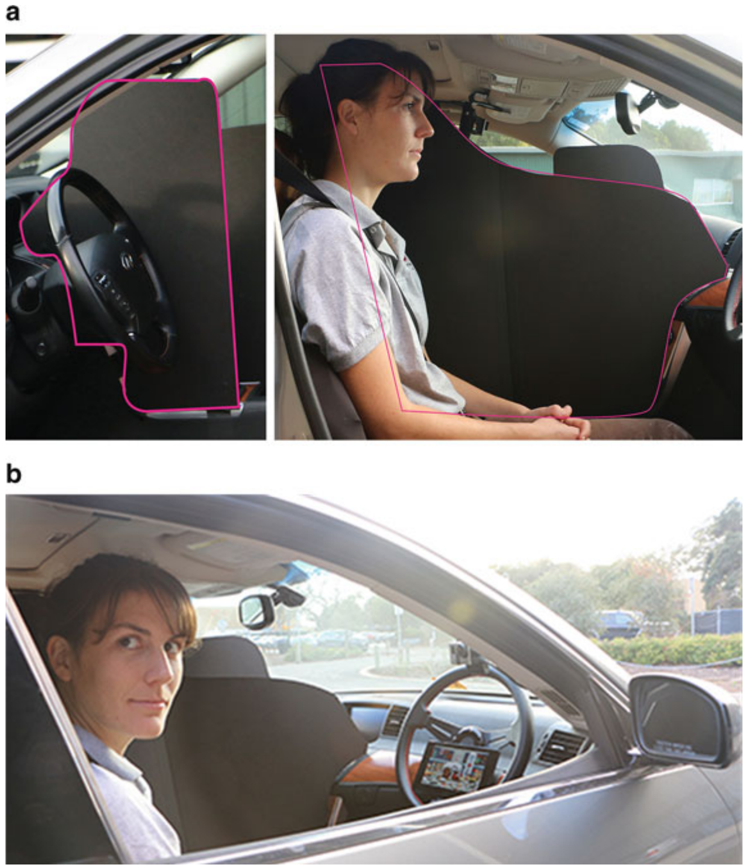


Fig. 5 (**a**) The Driving Wizard partition, as viewed from the driver's side (on the *left*), and passenger's side (on the *right*). (**b**) The participant's portion of the vehicle interior includes non-functional steering wheel, tablet interface, and video camera for recording gestural and facial reactions

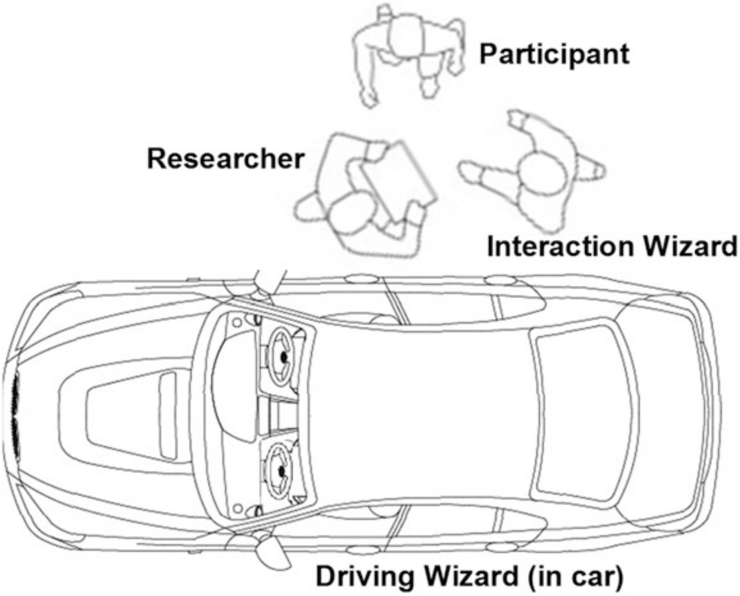


Fig. 6 Staging a participant's introduction to the RRADS vehicle. A researcher guides the participant into position; the Driving Wizard is already concealed behind the partition

On the Road

The vehicle follows a typically pre-selected course, which is predictable and safe, and which the Driving Wizard knows. Pedestrians, traffic lights, speed limit changes and high-density traffic can all be sources of opportunity or complication to study design. When the vehicle returns, the researcher opens the participant's door and engages him or her in light conversation, to allow the Driving Wizard the opportunity to drive away unseen.

Exit Interview

A qualitative exit interview provides an opportunity to uncover the salient points to the passenger's experience. Qualitative pilot phrases, such as "*How did the drive go, in general terms?*" can yield in-depth narrative responses which can be mined afterward.

2.2.4 Improvisation Sessions

The goal of our improvisation was to understand if pre-advising occupants of an autonomous vehicle's upcoming maneuvers—its starts, stops, lane changes or turns—might influence their trust in the system: effectively saying "*we're about* *to turn*" just before turning (or starting or stopping). We drove 35 participants along a course which included haptic (relating to touch) cues—expressed by the vibration or movement of the seatback or floorboard—to indicate these upcoming maneuvers.

The first prototype was a pneumatic base in the participant's foot well, just below his or her feet, that tilted in the direction the car was about to move. The second was an array of vibration motors, embedded in the seat back, which expressed patterns suggesting vehicle movement—such as a *cascade* from top to bottom to indicate an upcoming stop. The third was a pneumatic bladder placed behind the participants' shoulders which, when inflated on one side or the other, indicated that the vehicle was about to turn the opposite direction. We found that participants' guesses before the event were earliest in the case of the pneumatic floorboard, and latest in the case of the vibration array, suggesting that the floorboard provided the most effective early warning mechanism. We also found that environmental cues from the road, and from the vehicle's motion were always present, and played a significant role in participants' guesses.

The Hello Effect

At the start of the study, the vehicle greeted the participant using one of the prototype devices, or in the control case, with a revving of the engine. This was built into the study for practical reasons, to verify that the prototype was functioning properly, however, the interaction had an unintended effect on participants' experiences, with several cited the greeting as a source of comfort and a sign of amicability: "*I was surprised how much I trusted it. Even from the beginning, when it said, 'Hello,' it had enough of a personality. That one thing gave it enough personality for me to trust it,*" or "*When the car said, 'turn on,' or something, and then there was the air, it just kind of shot up, and I was like, 'okay, that's kind of interesting,' but it's like its way of communicating with you, rather than a voice thing.*"

Trust Through Driving Style

The driving style was, in itself, a form of improvisation. The Driving Wizard had to assume the role of an autonomous car and drive like one consistently in an unpredictable world. Road hazards, school and construction zones, and emergency vehicles required more attention. At these times, it was important that the Driving Wizard stay in character and keep the study going. The Wizard also had to prepare for changing lighting conditions, wear the right clothes to reduce noise while driving, and even hold in a sneeze every now and then.

Trust was high through all conditions. In fact, we did not find any significant statistical differences between conditions despite several of the pre-cuing systems (especially the floor boards) being very effective pre-cuing devices. We expect that this lack of difference is due to the driver's consistent and conservative driving style, so that the strength of this signal may have overwhelmed those from all other inputs.

Smooth Driving Is Safe Driving

During interviews, participants often referenced the car's safe driving style when questioned about trust. 30 % of responses to "*Did you trust the vehicle?*" strongly related to descriptions of smooth driving: "*The main thing was that it drove very smoothly*." This relevance also emerged when we asked participants to elaborate on why they trusted the vehicle: "*Because it was smooth and it wasn't too fast or jerky,*" or "*It wasn't anything sudden, or things that would normally make me go, oh my God, this is scary, stop*." The descriptions of smooth driving also related to descriptions of vehicle planning and awareness: "*Definitely smooth starts and stops. It sort of made it feel like the vehicle was planning what it was doing,*" or "*Something's a bit smoother, you realize that the person or the car kind of knows what's going on*."

Trust and Belief

The improvisation protocol did not employ deception. The partition separating the Wizard Driver from the participant was intended merely to help facilitate the illusion of an autonomous vehicle. And yet, about 25 % of participants believed that the system was fully autonomous. Another large portion of the participants believed the vehicle was partially autonomous and remotely controlled by the Interaction Wizard.

The prompt "*Did you trust the vehicle?*" was particularly helpful in uncovering how immersed participants became: "*I guess the computer was pretty cautious, which was pretty awesome. It was a much better driver than most humans that I know,*" or "*It made me feel like even though it wasn't a human, it wasn't of malicious intent*." A few participants who believed the system was autonomous revealed reservations about the technology: "*I just don't fully trust that car to drive on its own. Even though I had no bad experiences with this car, it just seems strange to me still and foreign to me that a car can drive itself*."

During more complicated maneuvers, some participants ascribed agency to the Interaction Wizard: "*There was a construction site. The guy was waving for me to move and I was like, I don't know what to do, so I was like, 'I really hope the car does something smart.' The car backed up and then the guy made more hand signals. I don't know if [the Interaction Wizard] or if the car did it*."

2.3 Ghost Driver

2.3.1 Introduction

With increasing capability of self-driving cars (Lari et al. 2014), each of a vehicle's occupants, including its operator, may become mere passengers, with no visible human driver. Long-established practices of communication between drivers and road users outside of the vehicle—such as making eye contact, nodding one's head or giving hand signs—may no longer be possible. Beside the arising concerns around safety, how comfortable might people feel walking or bicycling in front of autonomous cars if they do not receive acknowledgment that they have been noticed?

Prior work in social science, psychology and civil engineering has shown that a driver's gaze first goes to the face of a bicyclist (Walker and Brosnan 2007), and that pedestrians who stared at approaching drivers elicited greater stopping (Guéguen et al. 2015). Luoma and Peltola (2013) found that high speed was a signal from drivers that they did not intend to give way to pedestrians, and a similar field observation (Velde et al. 2005) showed that over half of pedestrians do not look for vehicles after arriving at a curb, but that all of them look at oncoming vehicles while crossing.

2.3.2 Prototype Systems

To explore these questions, we needed to evoke the impression that a car was driving autonomously, and in turn, deprive pedestrians and bicyclists of any chance to interact with a human (anywhere) in the car. While the California Department of Motor Vehicles issues licenses for testing autonomous technology, regulations require a human operator to occupy the driver seat at all times: to take over in an emergency, or for driving when autonomy is turned off (California 2015). Thus, a self-driving car would *have* to have a visible operator, and even though that operator might not be driving the car at the moment of an interaction, participants could wrongly interpret him or her as doing so.

We therefore developed a system (Rothenbücher et al. 2016) that would disguise both vehicle and operator, to make them appear to be autonomous to the outside world. For the car, we attached (that is, we did not connect and utilize) components from functioning autonomous cars—including a laser-based detector on the roof, radar units on the front corners, and cameras on the roof and dashboard—to an ordinary VW eGolf, and added vinyl stickers on the hood and doors saying *Stanford Autonomous Car* (Fig. 7).

For the driver, inspired by an invisible driver prank published on YouTube (Hossain 2013), we designed a *car seat costume*, to make him or her invisible to anyone outside of the vehicle (Fig. 8). The basic shape of the original seat was formed in wire mash, stabilized with paper-mâché, and covered with a regular seat cover. To give the driver peripheral vision, we covered the wire mash around the headrest (only) with sheer, see-through black fabric. The driver was dressed in black, including hands. Using only the bottom of the steering wheel, the driver could maneuver the car without being seen.

Fig. 7 The Ghost Driver car features autonomous vehicle props like a laser-based detector, radar units, interior and exterior cameras and decals on the hood and doors

Fig. 8 The seat cover costume includes two arm outlets so that the concealed driver could steer the car using the bottom of the steering wheel

2.3.3 Designed Behaviors

Implicit interaction theory (Ju 2015) suggests that pedestrian-autonomous vehicle interaction patterns at intersections might resemble the following: (a) a pedestrian approaches an intersection, (b) a car approaches the same intersection, (c) the pedestrian makes eye contact, (d) the driver makes eye contact, (e) the driver indicates not giving way, (f) the pedestrian waits, (g) the driver moves through the crosswalk, and finally (h) the pedestrian crosses—or alternatively: (e′) the driver indicates giving way, (f′) the driver stops and waits, (g′) the pedestrian crosses, and (h′) the driver moves through the crosswalk.

While this is our expected behavior, an autonomous car with no visible driver would likely break down the pattern in steps (c) and (d), critical points in which driver and pedestrian intent are communicated. This lack of nonverbal signals from a driver—or more correctly, the lack of a driver—is atypical, and serves to heighten awareness of the car, thereby making it appear to be more proactive. The transfer of agency from driver to car may lead to attempts to *repair* the interaction through improvised forms of iteration: repeatedly searching for a driver, staring at the location where the driver would be, trying to verbally engage a driver.

2.3.4 Improvisation Sessions

We developed a *breaching* experiment (Garfinkel 1991; Weiss et al. 2008), wherein we placed our mock autonomous car in a natural setting (Zhuang and Wu 2011), to observe how participants would respond. We video recorded interactions from multiple perspectives (on top of the car, on the street) (Millen 2000; Crabtree 2004), analyzed the footage, looking for behavior patterns and responses, and asked participants open-ended questions about their reactions and if they believed that the car was driving autonomously.

We held sessions in two locations: a parking lot with a pedestrian crosswalk at its exit leading onto a street, and a traffic circle with a large ratio of bicyclists. Both locations are highly frequented, especially between lectures and during lunch hour, so we ran sessions over 3 days from 11 am to 2 pm. At the crosswalk, the car waited in the parking lot, facing the exit and barely visible from the sidewalk. As soon as a pedestrian approached intersection, the invisible driver accelerated to arrive at the moment that the pedestrian was about to cross the street. We varied driving style from conservative on the first day to more aggressive at the second day, with the car approaching at a higher speed and stopping later. To create ambiguity, the car also briefly lurched forward after it had come to a full stop just as the pedestrian was about to cross. On the third day (a week later), we moved to the traffic circle, where the car entered and did a few circles before exiting again.

We employed a team of five researchers: a driver, a coordinator giving instructions to the driver and shooing away participants who came too close or lingered too long, and three interviewers, who asked participants questions just following their interactions. We recorded 67 interactions (49 at the crosswalk and 18 at the traffic circle) and interviewed 30 participants.

Pedestrians' Interaction Behaviors

Most people who interacted with the car noticed the missing driver (80 %) and believed that it was driving on its own (87 %). Also, bystanders were excited to see a car that appeared to be self-driving: taking photos or videos, talking about it with friends, even tweeting about it. These observations, and self-reported impressions on our questionnaire, show that our Wizard-of-Oz approach worked well and achieved its purpose.

The props and decals drew attention to the car, so that most participants stared a while to focus on it. Part of this focus was to look at the driver, which explains why so many saw that there was none, making it more difficult for them to predict what would happen next.

And yet, the video revealed that people were not overly shy about walking in front of the car. Although they assumed that it was self-driving, their behavior proceeded in the usual way. Of the 49 crosswalk interactions, only two people clearly tried to avoid passing in front of the car by walking around it, both on the second day when the car restarted after having come to a full stop. One said later "*I waited for a while to see what it's going to do, then tried to cross, but then while I was trying to cross, it attempted to start, so I stopped and waited.*" More often, we saw moments of hesitation, like stopping short or slowing walking pace. Sometimes, we saw people walk with greater deliberation and expressive motion, as if to explicitly signal that *a person is walking here* to the car (Fig. 9). So, particularly when the car stopped and restarted, established crossing procedures broke down, and participants' demand for interaction increased to resolve the ambiguity (Ju 2015)—only in this case, there was no driver to communicate and align with.

Forgiving the Newbie

Although the car sometimes misbehaved, nearly all people seemed to be tolerant and forgiving. We saw just one person on the second day who seemed to be upset by the car creeping into the sidewalk. Another person said "*I guess, if it were a person I'd have a really negative reaction towards them, but then, the autonomous car is a really interesting concept, so it was less negatively impacted.*"

Overall, people seem to have lower expectations than they would for human drivers, grant that the car is still learning, and admit that mistakes are part of that learning process. But at the same time, their expectations seem to be higher, since the technology is meant to eliminate human error. One participant, upon walking in front of the car, said "*the risk I took by crossing the intersection was higher than I realized, because nobody is behind the wheel of the car. At the same time, there are no human errors, there are just car sensors.*" Even though participants reported

Fig. 9 Participants walk in front of Ghost Driver, looking for a driver (on the *left*) and exaggerating walking behavior (on the *right*)

liking and trusting the car, some mentioned a certain unease, where they "*didn't feel very comfortable,*" "*wanted to make sure that it wasn't going to hit me*" or "*kept an eye out while crossing*." But what seems contradictory might just be a dual concept of trust: one which is spontaneous, in-the-moment and action-driven, and another which is more conceptual, and derived from mental models built over time about technology.

Compliance and Acknowledgment

There are two distinguishable elements of the interaction between pedestrians or bicyclists with a driver: one is to evoke *compliance* by connecting with the driver, and the other is to get *acknowledgment* that one was noticed. The first becomes irrelevant when the driver is a robot, in that people do not assume that the robot responds in a moody, impulsive or brash way. In other words, we do not need to give robots *the look*. Acknowledgment, however, remains relevant. People expect recognition that they have been seen, and if there is no driver, they look for it from the car, through its movement and behavior. This provides a design opportunity: to leverage people's knowledge of car behavior, such as expressive movement (like when the car's front *dives* as it stops at an intersection, or its rear *squats* as it accelerates away), lighting, or even sound, as explicit cues of the car's intentions.

3 Conclusion

In this paper, we have illustrated how the design thinking research techniques of *embodied design improvisation* have been applied to discover how interactions with autonomous vehicles should be designed.

In the WoZ sessions, participants wanted to share control with the car without assuming full control, and wanted to know exactly when the mode switch occurred. The car's delays in response and unperformed requests were acceptable, as long as it provided technical explanations. Each of these influenced participants' sense of trust, and the acceptability of the car as a conversation partner.

The RRADS platform pointed to influences that might be greater levers in trust than pre-cueing. The *Hello Effect* seems to indicate that an autonomous vehicle's perceived personality and driving style may be incredibly strong and salient factors in building users' trust in the system. In addition, participants seemed to be actively evaluating the vehicle's competence when it encountered complex situations, and its apparent ability to interact with construction vehicles or bicyclists seemed to reassure those who were initially skeptical of autonomous driving technology.

For Ghost Driver, further research will focus on the question of how important eye contact is for safety, once compliance is no longer an issue. In other words, is signaling acknowledgment a safety, or rather a convenience, feature of the car? We assume that this question is crucial for the design of such a signaling system because it defines the real needs of pedestrians and other human road users.

These three case studies exemplify how design thinking can help us to understand how people will respond to technologies that do not yet exist, and thereby help us to steer the direction of future technologies towards interactions that are safe and desirable.

References

Baum L (1900) The wonderful wizard of Oz (Books of Wonder). HarperCollins, Richmond, TX

Baltodano S, Sibi S, Martelaro N, Gowda N, Ju W (2015) The RRADS platform: a real road autonomous driving simulator. In: Proceedings of the International Conference on Automotive User Interfaces and Interactive Vehicular Applications, ACM, pp 281–288

California Department of Motor Vehicles (2015) Autonomous vehicles in California. http://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing/. Accessed Mar 2016

Crabtree A (2004) Design in the absence of practice: breaching experiments. In: Proceedings of Conference on Designing Interactive Systems, ACM, pp 59–68

Cross N (1977) The automated architect. Viking Penguin, New York

Dahlbäck N, Jönsson A, Ahrenberg L (1993) Wizard of Oz studies—why and how. Knowl-Based Syst 6(4):258–266

Garfinkel H (1991) Studies in ethnomethodology. Polity Press, Cambridge, p 36

Guéguen N, Meineri S, Eyssartier C (2015) A pedestrian's stare and drivers' stopping behavior: a field experiment at the pedestrian crossing. Saf Sci 75:87–89

Hoffman G, Ju W (2014) Designing robots with movement in mind. J Hum-Robot Interact 3 (1):89–122

Hossain R (2013) Drive thru invisible driver prank. http://youtube.com/watch?v=xVrJ8DxECbg. Accessed Mar 2016

Inagaki T (2009) Human-machine collaboration for safety and comfort. In: Proceedings of the ENRI International Workshop on ATM/CNS, Tokyo, Japan

Ju W (2015) The design of implicit interactions. Morgan & Claypool, San Rafael, CA

Kelley J (1983) An empirical methodology for writing user-friendly natural language computer applications. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, ACM, pp 193–196

Lari A, Douma F, Onyiah I (2014) Self-driving vehicles: current status of autonomous vehicle development and Minnesota policy implications. Preliminary white paper, University of Minnesota, pp 1–20
Luoma J, Peltola H (2013) Does facing traffic improve pedestrian safety? Accid Anal Prev 50:1207–1210
Millen D (2000) Rapid ethnography: time deepening strategies for HCI field research. In: Proceedings of Conference on Designing Interactive Systems, Processes, Practices, Methods and Techniques, ACM, pp 280–286
Mok B, Sirkin D, Sibi S, Miller D, Ju W (2015) Understanding driver-automated vehicle interactions through Wizard of Oz design improvisation. In: Proceedings of the International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp 386–392
National Highway Traffic Safety Administration (2013) Preliminary statement of policy concerning automated vehicles. US Department of Transportation, Washington, DC
Rothenbücher D, Li J, Sirkin D, Mok B, Ju W (2016) A field study investigating the interaction between pedestrians and driverless vehicles. In: Proceedings of the International Symposium on Robot and Human Interactive Communication, IEEE
Sirkin D, Ju W (2015) Using embodied design research as a design research tool. In: Proceedings of the International Conference on Human Behavior in Design, Design Society, pp 1–7
Sirkin D, Fischer K, Jensen L, Ju W (2016) Eliciting conversation in robot vehicle interactions. In: Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium Series: Enabling Computing Research in Socially Intelligent Human Robot Interaction, pp 164–171
Talone A, Fincannon T, Schuster D, Jentsch F, Hudson I. (2013) Comparing physical and virtual simulation use in UGV research: lessons learned from HRI research with two test beds. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting 57(1), 2017–2021
Velde A, van der Kamp J, Balera J, Savelsbergh G (2005) Visual timing and adaptive behavior in a road-crossing simulation study. Accid Anal Prev 37(3):399–406
Walker I, Brosnan M (2007) Drivers' gaze fixations during judgements about a bicyclist's intentions. Transport Res F: Traffic Psychol Behav 10(2):90–98
Weiss A, Bernhaupt R, Tscheligi M, Wollherr D, Kühnlenz K, Buss M (2008) A methodological variation for acceptance evaluation of human-robot interaction in public places. In: Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, IEEE, pp 713–718
Zhuang X, Wu C (2011) Pedestrians' crossing behaviors and safety at unmarked roadway in China. Accid Anal Prev 43(6):1927–1936

Part III
Prototyping

Can Anyone Make a Smart Device?: Evaluating the Usability of a Prototyping Toolkit for Creative Computing

Joel Sadler, Lauren Aquino Shluzas, Paulo Blikstein, and Sakti Srivastava

Abstract Can anyone make a smart device? Affordable sensors, actuators and novice microcomputer toolkits are the building blocks of the field we refer to as *Creative Computing*. With the growing maker movement, more tools are becoming available to novices, but there is little research into the usability evaluation of these toolkits. In this chapter, we discuss the importance of closing the gap between idea and prototype, the need for systematically evaluating the usability of novice toolkits, and a strategy for doing so. Specifically, the chapter presents the Tiny Device Test, a method for evaluating the usability of novice electronics toolkits. Using a standard set of building challenges based on common household electronics, we discuss methods for evaluating the Bloctopus toolkit, which was designed for novice electronics prototyping with low-resolution materials. This work aims to contribute to the idea of "making simple things simple, and complex things possible," with prototyping toolkits of the future.

J. Sadler, Ph.D. (✉)
Center for Design Research, Building 560, 424 Panama Mall, Stanford, CA 94305-2232, USA

School of Education, Learning Sciences and Technology Design Program, 520 Galvaz Mall, CERAS Building, Stanford, CA 94305, USA
e-mail: jsadler@stanford.edu

L.A. Shluzas, Ph.D.
Center for Design Research, Building 560, 424 Panama Mall, Stanford, CA 94305-2232, USA

P. Blikstein, Ph.D.
School of Education, Learning Sciences and Technology Design Program, 520 Galvaz Mall, CERAS Building, Stanford, CA 94305, USA

Department of Computer Sciences, School of Engineering, 353 Serra Mall, Stanford, CA 94305, USA

S. Srivastava, M.B.B.S., M.S.
Division of Clinical Anatomy, Stanford University, 269 Campus Drive, CCSR Building, Room 0125, Stanford, CA 94305, USA

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_10

1 Motivation: Closing the Gap Between Idea and Prototype

Prototyping is a core element of the design process, where we create early forms of ideas as a way to test the essence of a design (Houde and Hill 1997). These rough prototypes may sometimes be simple napkin sketches, or 60-s cardboard creations held together by tape. However, when prototyping electronic devices, technical novices may not have the knowledge necessary to create prototypes that function. Prototyping with electronic circuits, sensors, actuators, and programmable microcontrollers presents significant technical barriers that can shift the focus away from creative problem solving, into technical debugging (Sadler et al. 2015). Although researchers have created many electronic toolkits to reduce the technical burden, there is little research into the specific factors that make a toolkit more usable as a prototyping tool. Norman (1986) describes this usability gap between idea and implementation, as the gulf of execution (Fig. 1). The motivation of this research is to better understand how we might close this gap between novice's ideas and prototypes in a systematic way.

> Imagine for a moment that you are a junior physiotherapist with an idea for a smart medical device that will help patients automatically track their recovery after a knee injury, with a wireless range-of-motion logger (Fig. 1). Let us call this idea a "Smart Goniometer". In additional to the physical design of a wearable form factor, you need some way to sense the angle of the knee, give feedback to the user (e.g. visual), store, transform and transmit the sensor signals. As a thought experiment: Do you feel confident that you could build this prototype? What tools would you use? How long would it take? What affordances would the tools need in order to enable a technical novice to prototype this device in under an hour?—Smart Goniometer Challenge, author.

This example highlights a critical observation about prototyping ideas inside the gulf of execution, namely that the **tools have a major impact on the electronics prototyping process.**

2 Need for Systematic Usability Evaluation of Novice Toolkits

To address some of the challenges with prototyping electronics, a wide variety of electronic toolkits have emerged over the last 30 years (Fig. 2). These toolkits range in their intended audience from children, artists, hobbyists, engineers and designers. They include (1) toy electronic construction sets, such as Lego Mindstorm's programmable bricks, (2) microcontroller platforms, such as the Arduino (Mellis et al. 2007) "breakout" boards, (3) modular block-based construction sets such as littleBits (Bdeir et al. 2009), and (4) affordable microcomputers such as the Raspberry Pi (Upton and Halfacree 2012). A comprehensive review of novice toolkits, and their tradeoff during prototyping, appears in Sadler (2016).

Fig. 1 The gap between idea and prototype for the "Smart Goniometer"

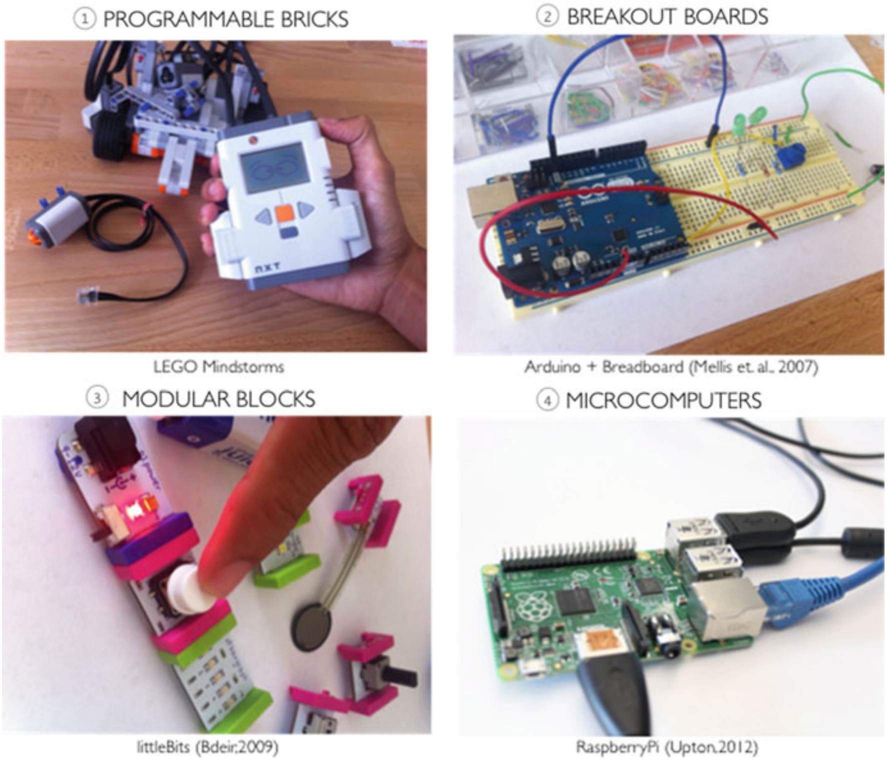


Fig. 2 Four common approaches to electronic toolkits, from bricks to blocks

Despite the variety of approaches, there have been few attempts to develop a systematic way to evaluate toolkit usability (Blikstein 2015). Platforms, such as Arduino, claim to be novice-friendly, but in practice can be intimidating for many novices without any electronics or programming experience. In a benchmark Arduino study, (N = 68) high school students were asked to make an LED blink in 10 min, but 47 % of participants had an error that required expert intervention (Sadler et al. 2016). The Raspberry Pi, which was originally designed for use in schools, has sold over seven million units over the last 5 years, but the majority of users are technical, rather than children (Cellan-Jones 2015). Are the popular toolkits of today truly meeting the needs of the average novice?

3 From Tiny Tasks to Big Ideas: Strategies for Usability Testing

Recalling that our goal is to enable novices to prototype smart devices: How do we develop a general method to test the usability of novice toolkits? It is difficult to predict exactly what device a novice will attempt to build, given the open-ended and ambiguous nature of design. However, smart devices are built out of a common set of building blocks: sensors, actuators and processors (Sadler 2016). Prototypes are therefore distinguished by the *unique* way that these building blocks are combined and coordinated towards solving a problem. Turning on and off an output (e.g. LED), reading discrete and continuous sensor voltages (e.g. motion sensor), and linking actions to conditional reactions (e.g. “if-this-then-that”), are generalized actions. If we can break down electronics prototyping into a set of common tasks, we can then begin to dissect usability at a task level. This focus, on “simple tasks first,” reflects lessons learned from analysis of task complexity in the field of User Centered Design (Norman 1986). Innovative ideas begin from simple prototypes, and complex prototypes are created in tiny steps.

We suggest the following strategies in developing a usability test for a given novice electronics toolkit:

1. **Define the domain**. Toolkits may be designed for general-purpose electronics prototyping, or may be specialized towards certain types of domains such as hand-held devices, robotic vehicles, medical devices or wearable sensing. The domains should be matched if comparing toolkits.
2. **Set a floor and ceiling of complexity, in a fixed timeframe.** What is the most complex device that we expect a novice to be able to build with the system in an hour? Conversely, what is the simplest possible device? In the spirit of rough prototyping we recommend a fixed 1-h time limit.
3. **Define a scaffolded set of simple devices**. Create a set of devices, with increasing complexity, that reflect the functionality floor and ceiling. For example: starting with “make a flashlight”, and ending with a device with multiple inputs and outputs linked by logic. Making the same set of devices across different toolkits can be used as a benchmark comparison.
4. **Observe novices building the set of devices**. Can novices build all the devices in the set timeframe? Measuring task-performance, direct qualitative observation, and surveying pre-post cognitive changes such as self-efficacy change, are all critical measures (Sadler et al. 2016).
5. **Provide an open-ended design challenge**. In addition to a fixed set of tasks, providing an open challenge reveals how novices will play with the toolkit in free-form, and under more realistic design conditions.

6. **Have experts use the system**. If experts have difficultly completing the same tasks in time limit, this may be a strong signal of usability weaknesses.

The guidelines presented above were developed with the observation that popular toolkits, such as Arduino, often require more time, tools and knowledge, than can be compressed into a 1-h time prototyping frame with true beginners. Novices may indeed have more time than 1 h to learn the technical intricacies of a system; but if there are large usability gaps, such as high error rates on elementary blinking LED tasks, these may reflect frequent barriers to the creative flow (Sadler et al. 2016). The development of a usability benchmark is one way to (1) ensure a minimum level of novice task performance and (2) highlight differences between toolkits.

4 The Bloctopus Novice Toolkit

This research proposes the Tiny Device tests as a method for testing electronics toolkits. We originally designed this evaluation method as a way to evaluate the *Bloctopus* novice toolkit (Fig. 3). *Bloctopus* is a modular block-based electronics set, aimed at enabling novices to prototype smart devices. Rather than fixed black-boxes, *Bloctopus* uses a hybrid design with modules that are both open to modification, but are initially connected by plug and play interaction over LEGO-shaped USB sticks. The toolkit combines these modules with a visual and textual programming over a web-based interface. Physically, the toolkit is designed to integrate with paper prototyping as well as LEGO blocks, and so is well suited for 1-h prototyping time frames proposed by the Tiny Device Test (Sect. 5). Full technical details of the system are presented in Sadler et al. (2015).

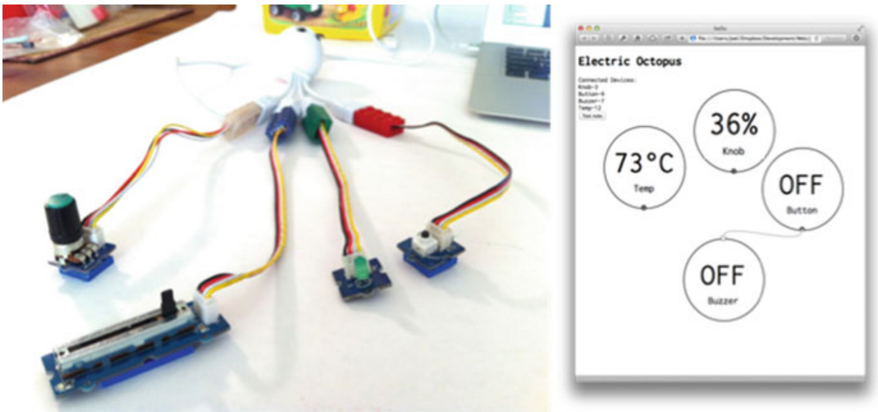


Fig. 3 The Bloctopus physical prototyping toolkit: An example of a modular electronics and software toolkit (Sadler et al. 2015)

5 The Tiny Device Test

How can we systematically evaluate an electronics toolkit for usability? Using the evaluations guidelines presented previously, we focus on the domain of smart device smart device prototyping at the scale of hand-held devices. The intended users of the toolkit are novices with no programming or electronics experience. Our goal was, in a **1-h time** frame, for novices to be able to:

I. Individually connect and read input/sensors, and control output/actuators.
II. Combine multiple sensors, actuators and logic into interactive devices.
III. Apply this to an open-ended design challenge without expert intervention.

The "Smart Goniometer" challenge, previously introduced, is an example of a typical device we would like novices to be able to prototype in a 1-h session. Following the engineering guidelines previously developed in the Stanford Paper Robot activity (Analytis et al. 2015), our functionality ceiling is: that novices should able to combine two or more sensors (e.g. angle sensor, button), with two or more outputs (e.g. sound buzzer, light), with conditional logic between sensors and outputs. There are two sections to the test (1) a set of eight common household devices in increasing complexity (2) an open-ended design challenge. The building tasks are scaffolded by difficulty, with each linked with a critical learning objective as follows (Fig. 4):

1. **Selecting Inputs/outputs**: *a discrete input can be mapped to an output.*
2. **Arbitrary mapping:** *any inputs can be mapped to outputs.*
3. **Conditional logic**: *actions can be triggered based on conditions.*
4. **Multiple inputs, outputs:** *multiple combinations can be connected.*

We determined that the learning objectives in (Fig. 4) represent a minimum set of knowledge needed for a novice to create a smart device, such as the smart goniometer. Device challenges may be substituted for other devices that achieve the same learning objective. We use common household devices to give the novice practice in combining components of the toolkit, with familiar interactions (Fig. 5). The eight challenges on the test are:

1. "**Make a doorbell**"—when a button is pressed, turn a buzzer on.
2. "**Make a flashlight**"—when a button is pressed, turn a light on.
3. "**Make a fan**"—as the knob is turned, make the fan spin faster.
4. "**Make a moodlight**"—as the knob is turned, make the light brighter
5. "**Make a thermostat**"—If the temperature is over 70 F turn on the heater.
6. "**Make a smart AC**"—if the temperature is under 70 F, turn on the fan.
7. "**Make a smart garage door**"—When the button is pressed and there is no motion in the garage then turn on the door motor.
8. "**Make a burglar alarm**"—When the room is dark, if there is any motion the room, then turn a buzzer on.

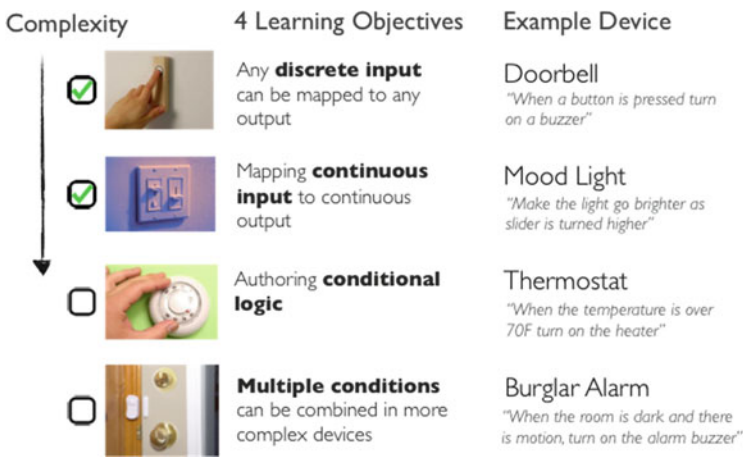


Fig. 4 Task scaffolding of the Tiny Device Test

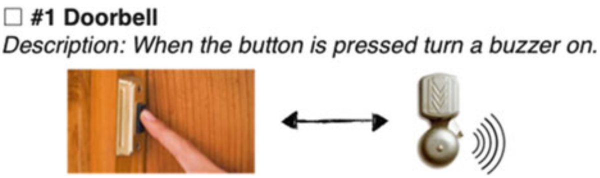


Fig. 5 An example tiny device challenge

5.1 *Challenge Structure*

Participants are given (1) the toolkit parts and assorted paper crafts (Fig. 6), (2) an initial prompt to set the creative mood, followed by (3) the eight household devices challenge, and (4) finally an open-ended prototyping task. The opening prompt is, inspired Marsh et al. (1996), introduces the challenge:

> Imagine a planet like earth existing somewhere else in another universe. Alien creatures have just landed on this planet and live in simple shelters. The creatures in this world look much like earth flamingos and love electronic devices. Your task is to design new electronic devices to solve one of the alien's electronic problems. To help you in this challenge you will first practice making some electronic devices that can be found in many homes on earth. After the practice session you will be asked to imagine and build possible creative ways to solve a design challenge for our alien friends.

With pre-post surveys, direct observation, and task completion measures, we can use the Tiny Device structure to evaluate a toolkit in a controlled study.

Fig. 6 Example of "tiny devices" made of out cardboard

6 Methods: Evaluating Bloctopus with the Tiny Device Test

Here we present an example of how the Tiny Device method has been adapted for the usability evaluation of the Bloctopus Toolkit. The primary measures in this evaluation include:

1. Task performance of the toolkit, which had two components:
   (a) **Task competition:** percentage of working tiny devices created.
   (b) **Design ranking score:** a measure of how closely novice's choices mirror those of experts.
2. Psychological Measures of:
   (a) **Self-efficacy change:** the confidence in one's ability to create electronic prototypes in the future. This was segmented by (1) electronics and (2) programming self-efficacy. (1–7, Likert)
   (b) **Creative feeling change:** How well the novice self-describes him or herself as feeling in a creative state of mind. (1–7, Likert)

6.1 Groups

We recruited three groups to participate in the study with (N = 40). Both experts and novices were used in this study in order to compare differences. (N = 20) participants were given the tiny device test with the physical components. Novices (N = 13) were defined as anyone without formal experience in programming or electronics. Experts (N = 7) were recruited from mechanical, electrical and computer science departments at Stanford University. Novices were recruited from non-engineering departments. As a control group, (N = 20) participants were recruited on Mechanical Turk, but without physically building components—the control group provided only written ideas (Fig. 7).

6.2 Hypothesis

The authors hypothesize that using the toolkit, in combination with the test:

H1. Both novices and experts will create all working prototypes in under an hour. (100 % task completion).
H2. Novices will have greater increases in self-efficacy compared with experts and the control.
H3. Novices will have design ranking scores similar to experts, compared with the control.

These hypotheses and metrics build on previous studies that examine self-efficacy and task-performance as primary measures of usability (Sadler et al. 2016; Blikstein 2015, 2016). The structure of the study follows the same pre and post survey instruments as Sadler et al. (2016).

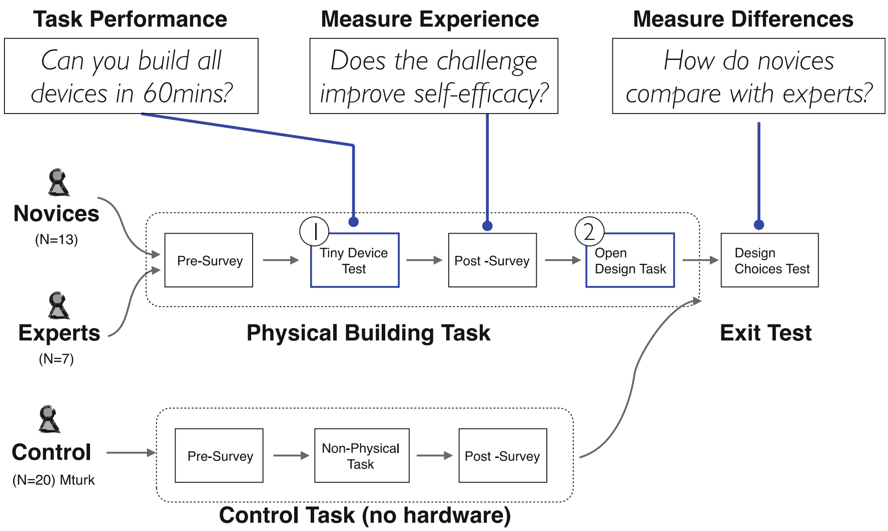


Fig. 7 Study design

6.3 Open-Ended Design Challenge

As previously described, the first half of the test requires prototyping a standard set of eight common house devices. The second half of the test is meant to give novices an opportunity to exercise creative problem solving, while also trying to build working prototypes. Participants were given a prompt, as well as 8 min of sketching time, before being asked to physically prototype their strongest idea. The challenge used for this study was an adaptation of an unsolved problem in the surgical field, where surgeons often lack ergonomic feedback during extended hours in the operating room, and may shift their weight to one leg for long time periods. We disguised the clinical design challenge as an alien creature, in order to increase the playfulness of the challenge (Fig. 8).

6.4 Exit Test: Design Ranking Score

As a final metric we wanted to gain some insight into how similar novice choices were compared with those of experts. The assumption is that experts will tend to make stronger design choices than novices, and so comparing discrete design choices gives one measure of performance. Participants were given a fictitious prototype description of a smart shoe, with one sensor for detecting positions, and one output for giving feedback. As an exit survey, participants were asked to rank order the three inputs and three outputs that would work best for the design challenge. Unbeknown to the participants, some choices were deliberately included

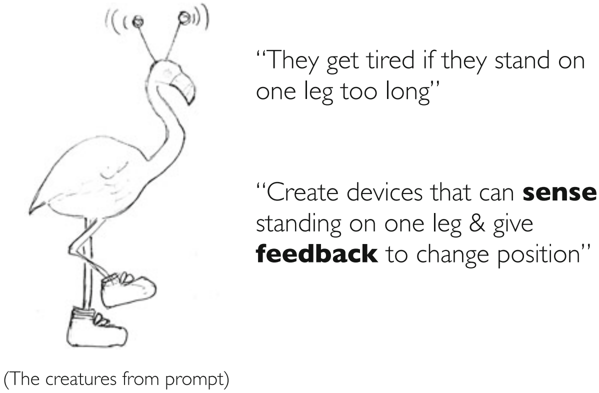


Fig. 8 The open-ended design prompt based on surgeon fatigue problem

as poor design choices based on the prompt. For example, participants were explicitly told that the end-users operated in a noisy environment. Experts who are more familiar with the components are less likely to select the sound buzzer as a feedback output. Similarly using a light sensor to detect shoe position is less common with experts since they are aware of the fact that the method will not work well in all ambient lighting conditions (Fig. 9). Using expert interviews to calibrate the design choices, we calculated a weighted rank sum of scores for each participants exit survey.

6.5 Analysis

Task performance % was calculated by coding a binary completion score for each task. Participants were asked to demonstrate each prototype functioning according to the description (e.g. "when the button is pressed turn the LED on"). Partially complete prototypes were coded as incomplete. For the pre-post surveys, the differences between each group's pre-post self-efficacy and self-reported creative feeling was calculated, and statistically analyzed for significance using one-way ANOVA. Pairwise tests for significance between control group and novice and expert groups, were conducted with a two-tailed t-test. [The results of this analysis will be published in future publications] (Fig. 10).

7 Discussion

7.1 From "Big O" to "Big U" (for Usability)

The Tiny Device Test is just one start at a proposed minimum usability benchmark. There were many limitations to a study of this kind, including: (1) the partial

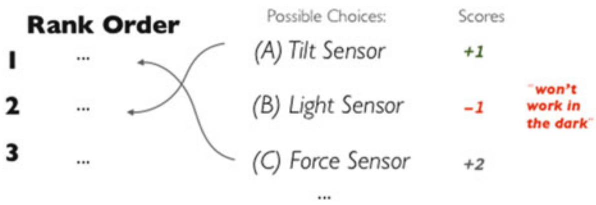


Fig. 9 The open-ended design prompt based on surgeon fatigue problem

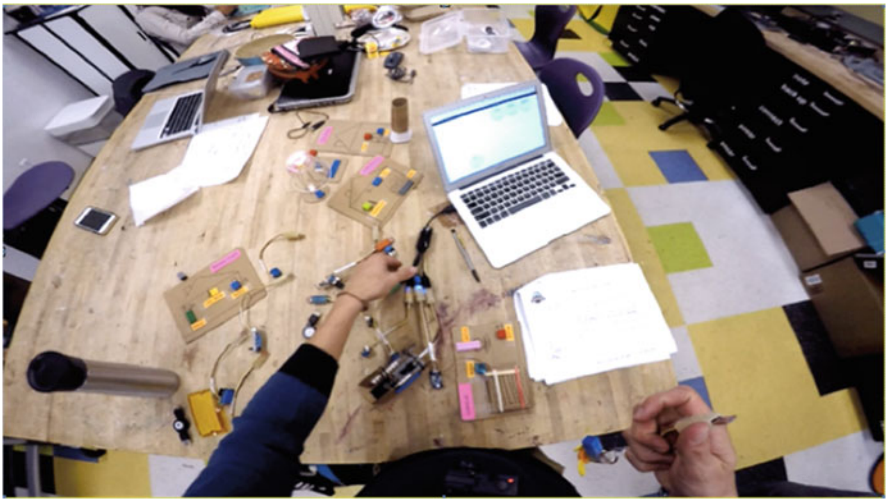


Fig. 10 First-person view of participant during the tiny device test (Head-mounted GOPRO camera)

reliance on self-reported measures, (2) limited sample size, and (3) a non-homogenous test group. The functionality ceiling in the test (a device with two inputs, two outputs, and conditional logic) is not representative of all prototypes. As creative computing tools become cheaper and more powerful, we expect that we will need to redefine our expectations of the functionality ceiling. However, as a tool to determine a minimum threshold, these simple tasks may remain applicable over time. Decades of programming languages have used the task to print "Hello World," as a long-lasting first benchmark in comparing programming languages. We hope that the Tiny Device Test encourages future designers of toolkits to apply the "Hello World" tests for electronics prototyping.

In Computer Science, the idea of a rough benchmark for a comparison is familiar to algorithm designers. Donald Knuth popularized the idea of a standard "Big O" notation, where we could quickly evaluate the differences between different algorithms by isolating the largest limiting factor of an algorithm's speed as the number of elements, n, grows (e.g. n, n^2, n^3) (Knuth 1976). The beauty of Big O notation was that it gave algorithm designers a way to quickly evaluate the worst-case performance as complexity increases. We propose an analogy to electronics toolkit prototyping, where there is a defined task complexity spread over a number of small input tasks. Our goal here is not more powerful algorithms, but rather more powerful tools, and in fewer steps. Can we move into an era where we instead evaluate a Big U for usability?

8 Conclusions

Our goal, as the designers of new prototyping tools, is to enable anyone to be able to prototype their ideas. However the gulf of execution, between ideas and prototypes, depends on the usability of the tools. There is little research into the evaluation of toolkits, and how usable they are for technical novices. To remedy this, we presented the design of the Tiny Device Test, a method to evaluate the usability of novice electronics toolkits. This evaluation method uses a standard set of building challenges based on common household electronics. Using this method we evaluated the usability of the Bloctopus toolkit, which was specifically designed for novice prototyping. We hope that this research contributes to the idea of "making simple things simple, and complex things possible," with the prototyping toolkits of the future.

References

Analytis S, Sadler JA, Cutkosky MR (2015) Creating paper robots increases designers' confidence to prototype with microcontrollers and electronics. Int J Des Creativ Innovat 1–12. Chicago. doi:10.1080/21650349.2015.1092397

Bdeir A (2009) Electronics as material: littleBits. In: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, ACM, pp 397–400

Blikstein P (2015) Computationally enhanced toolkits for children: historical review and a framework for future design. Found Trends Comput Sci 9(1):1–68

Cellan-Jones R (2015) Shrinking to zero: the Raspberry Pi gets smaller. BBC News. http://www.bbc.com/news/technology-34922561, 26 Nov

Houde S, Hill C (1997) What do prototypes prototype? In: Handbook of human-computer interaction, vol 2. Elsevier, Cambridge, pp 367–381

Knuth DE (1976) Big omicron and big omega and big theta. ACM Sigact News 8(2):18–24

Marsh RL, Landau JD, Hicks JL (1996) How examples may (and may not) constrain creativity. Mem Cogn 24(5):669–680

Mellis DA, Banzi M, Cuartielles D, Igoe TA (2007) An open electronics prototyping platform. In: Proceedings of the Conference on Human Factors in Computing (alt.chi), ACM

Norman DA (1986) Cognitive engineering. In: Norman DA, Draper SW (eds) User-centered system design: New perspectives on humancomputer-interaction. Erlbaum, Hillsdale, NJ, pp 31–61

Sadler J, Durfee K, Shluzas L, Blikstein P (2015, January) Bloctopus: a novice modular sensor system for playful prototyping. In: Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction, ACM, pp 347–354

Sadler J (2016) The anatomy of creative computer: enabling novices to prototype smart devices. Doctoral dissertation, Stanford University

Sadler J, Shluzas L, Blikstein P (2016) Building blocks in creative computing: modularity increases the probability of prototyping novel ideas. Int J Des Creativ Innovat 1–17. doi:10.1080/21650349.2015.1136796

Upton E, Halfacree G (2012) Meet the Raspberry Pi. Wiley, Hoboken, NJ

Making Examples Tangible: Tool Building for Program Comprehension

Marcel Taeumel and Robert Hirschfeld

Abstract Best practices in design thinking suggest creating and working with tangible prototypes. In software engineering, programmers interact with source code more than with customers. Their intent is to understand the effects of abstract source code on programs in execution. Existing tools for program exploration, however, are tailored to general programming language concepts instead of domain-specific characteristics and programmer's system knowledge. In this chapter, we establish the need for adapting programming tools in use when *navigating*, *viewing*, and *collecting* examples to increase tangibility, that is, clarity of a concept or idea based on what can be experienced on screen. We present our Vivide tool-building environment, which is a data-driven, scriptable approach to constructing graphical tools with low effort. By exploring common programming scenarios, we conclude that tool building does not have to be a detached, effortful activity but can be accomplished by the same programmers who detect deficiencies during their programming tasks. Then exemplary information about software systems can become tangible.

1 Introduction

Best practices in design thinking include *prototyping*, which helps verify assumptions and gain a better understanding of the often abstract and unclear problem and solution space. Such prototyping activities will typically produce *tangible artifacts*. This means that designers work with real-world materials such as paper, glue, pencils, whiteboards, and index cards. Externalizing thoughts and ideas in simple but concrete things can foster team communication or enable first user testing. The quality of prototypes can range from low-end to high-end while retaining their

M. Taeumel (✉) • R. Hirschfeld
Software Architecture Group, Hasso Plattner Institute, University of Potsdam, 14482 Potsdam, Germany
e-mail: marcel.taeumel@hpi.de; robert.hirschfeld@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_11

exemplary nature: It's not about experiencing the final product but about holding in your hands a low-cost, incomplete, yet tangible analogy.

When the product is going to be a piece of software, prototyping can support programmers and customers to talk about requirements in a shared language. Typical tangible artifacts include user stories (Cohn 2004) on index cards, bricolages of graphical user interfaces (Shneiderman and Plaisant 2010), or descriptive personas (Courage and Baxter 2005) on whiteboards.

Despite the many social aspects involved in software engineering, most of the time programmers have to focus on *talking to source code* instead of customers. Among all information related to software systems, only the source code is always up-to-date because it describes the system's actual behavior. Every programming activity includes reading and modifying source code. There is in fact an overwhelming amount of information available in large systems. Not only the numerous lines of source code but also related artifacts, such as program execution traces and external documentation, can support understanding. Programmers continuously ask questions about system parts while fixing bugs or adding features. The helpful answers to these questions represent *tangible examples* of information needed to accomplish programming tasks.

In programming-the notion of tangibility does not address primarily a physical representation but an aspect of cognition. Examples representing concepts, mechanisms, or intents should be "capable of being precisely identified or realized by the mind"[1] to be tangible. In a given task, programmers have to understand the particular mapping between the *problem domain* and the *source code* as well as between the source code and the software system in *execution*. General questions include: "How is domain knowledge represented in source code?" and "How are the rather abstract descriptions from the source code put into action during program execution?" as depicted in Fig. 1.

To acquire this understanding, programmers use *tools* for *collecting*, *navigating*, and *viewing* source code and other related software artifacts. These tools are called browsers, editors, debuggers, or explorers. Once created without anticipating specific domains, these tools provide *only generic support* for the underlying programming language constructs. For example, the language Smalltalk knows packages, classes, categories, and methods. Tools support finding those artifacts and navigating their relationships. There are common strategies to approach a larger system such as "top-down" where programmers begin with the most abstract or coarse-grained structure and then dive into the details (Von Mayrhauser and Vans 1995).

However, program comprehension is a creative activity influenced by human factors and domain-specific characteristics. There is typically no single strategy that works equally well for all programmers or in all known domains. On the one hand, people vary in terms of existing knowledge, cognitive capabilities, mood and motivational triggers (Csikszentmihalyi 2008), or even vision. On the other hand, domains carry specific terms and rules that have to be considered when using

[1] Definition of "tangible" from http://merriam-webster.com, accessed on Dec 3, 2015.

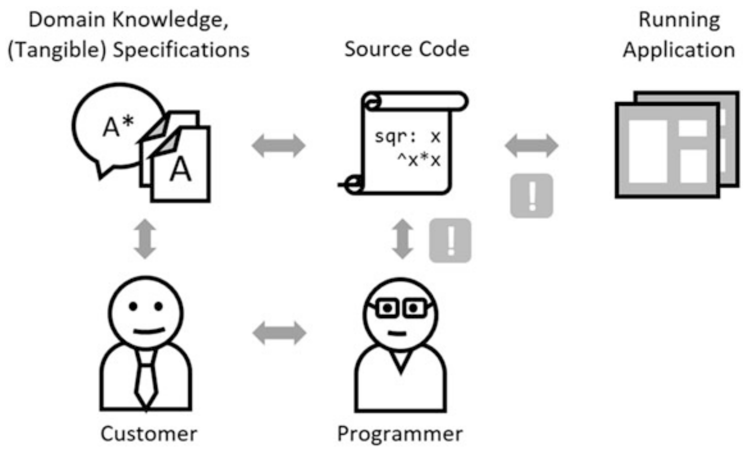


Fig. 1 Programmers write source code to make applications fit customer specifications. Large software systems pose challenges in uncovering the mapping between domain and code as well as code and running applications

standard tools. For example, to answer "Why doesn't this Tetromino[2] rotate clockwise when the arrow-right key is pressed?" requires mapping from specific terms to generic actions, which the tools dictate in this case as "Add breakpoint", "Browse declaration", and "Open documentation".

By accommodating personal preferences and domain-specific characteristics, programmers are expected to find tangible examples more efficiently. Many tools offer a *means of configuration* by switching color schemes, keyboard shortcuts, window layouts, content filters, or integration with other tools. This possibility can reduce the number of user interactions and the chance to make mistakes due to cognitive overload or slip ups (Norman 2002). Hence, it can save time. As tools being software systems themselves, programmers have the skills to approach any task from the best possible angle.

However, there are two serious challenges that affect using these skills: (1) Not many programmers reflect regularly about their working habits yet come up with ideas for improvement and (2) simple tool customization is quite limited, extended customization typically not worth the usually high effort. At the end of the day, current habits remain unchanged. Programmers keep on using generic tools for a specific information space. This habit impedes navigating, viewing, and collecting software artifacts and hence finding the tangible examples that quickly answer program comprehension questions.

We think that programming languages and programming environments influence the way programmers think about tooling and the possibilities for improving their workings. In this chapter, we describe several existing means and triggers that

[2] A "Tetromino" is the block in Tetris games. In such games the player has to arrange falling pieces of varying shapes to fill rows to gather points.

foster the creative nature of program comprehension to gather the tangible examples more efficiently to answer the questions that arise in programming tasks. We emphasize the benefits of having a self-sustaining, reflective environment with access to the tool's underlying source code. We present our tool building environment, called Vivide.[3] It is implemented in Squeak/Smalltalk[4] and employs a data-driven perspective on programming tools. It supports a scriptable way to create graphical tools for programming with low effort.

In Sect. 2, we give background information and motivate the need for employing the best practices of self-sustaining programming environments such as Squeak/Smalltalk. We present several examples of deficiencies in generic programming tools in Sect. 3, which reveal their deficiencies when applied to domain-specific programming tasks. We explain and apply our Vivide programming and tool building environment in Sect. 4. Finally, we conclude our thoughts in Sect. 5.

2 Learn About Your Environment's Possibilities

In this section, we establish the need for learning and applying the concepts that particular programming languages and environments provide. We argue that this is a requirement for efficient reflection about personal working habits and the adaptation of programming tools in use.

2.1 Being Aware of Different Concepts

Creativity draws from existing knowledge and previous experiences. Programmers can be creative when developing a program comprehension strategy, especially if they know and understand their surroundings, that is their programming languages, tools, and environments used. Programming tools are also described with source code, just like the application that has to be created for a customer. Hence, there is the chance that programmers can understand how tools work and how they can be modified to better support the circumstances. Unlike many users of software systems, programmers train particular skills that allow them to understand the building blocks of programs and their algorithmic, logic nature. Unfortunately, there are also many programmers who treat programming tools as "black boxes" and hence remain simply users. They are, however, unwilling to dig into internals and improve the modus operandi.

The tools' sources have to be available (Weiser 1987). The curiosity of a programmer is not worth a dime, if there is no access to a human-readable

[3]The Vivide environment: http://www.github.com/hpi-swa/vivide, accessed on Dec 3, 2015.

[4]The Squeak/Smalltalk programming system: http://www.squeak.org, accessed on Dec 3, 2015.

description of the program. Source code gets translated into byte code to be interpreted by a virtual machine or compiled into machine code to be executed on the actual computer hardware. Such target representations are hardly readable by programmers. The preservation of the source code is required for maintenance. Having originated in a commercial context, many programming tools (or environments, respectively) such as Eclipse and Visual Studio do not offer sources. There are, however, full-source environments such as Squeak/Smalltalk where applications and tools are open, readable, and ready to be modified.

Whenever programmers learn new languages or tools, they build on existing knowledge and try to apply familiar concepts. This works quite well because many new ideas originate from previous experiences and retain best practices. For *imperative* programming languages, this might be the for-loop or if-else-conditional expression. For programming tools, this usually includes having text editors, copying code via the keyboard shortcut **[Ctrl]** + **[C]** and pasting contents via **[Ctrl]** + **[V]**, or setting breakpoints and invoking a debugger. However, programmers have to be open for new ideas. Especially when switching from a familiar language or environment to an unfamiliar one. That other environment might be used in familiar ways but its power can only unfold once its unique concepts become clear. For a brief example, in Squeak/Smalltalk, programmers can modify running applications by easily exchanging portions of code. However, it is also possible to kill and restart applications over and over again after every little modification. If programmers fail to learn and apply the concepts, patterns, and idioms, they cannot improve their working habits and hence only work inefficiently in the given environment.

2.2 Tool Mechanics

In this report, we focus on *graphical tools for programming*. These are tools that have windows, buttons, lists, text fields, or other kinds of interactive widgets. We think that programmers can benefit from graphics-based interfaces in terms of increased information density and convenient input methods such as mouse and touch. Text-based interfaces, for example command lines, are still popular in several communities and maybe one indication for inconvenient designs in the graphical world. However, this is precisely where programmers can take the opportunity to tailor their tools as needed. This can work if the mechanics of the underlying tool-building framework are comprehensive and easy to apply.

There are many ways to model the structure of programming tools. We think that it is useful to distinguish between the data that is accessed and the visuals that are produced as depicted in Fig. 2. For tool builders, a *query language* is used to access the data. For tool users, a *presentation language* has to be learned to make sense of the visuals. Usually, there is also a *mapping language* because many software artifacts do not have an inherent graphical representation and hence have to be mapped to the graphical properties of standard widgets such as scrollable lists or text boxes.

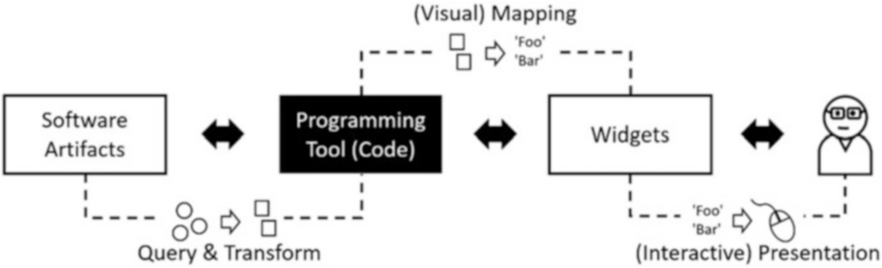


Fig. 2 Graphical tools for programming query system data to retrieve software artifacts such as source code, external documentation, and run-time traces. A subset of the artifacts' information is extracted and mapped to what interactive widgets support such as textual labels and color properties. The tool's source code is basically an adapter between databases and widgets

Presentation languages encompass interactive, graphical widgets. For example, standard tools offer buttons, lists, trees, or tables. Sometimes they have maps or charts to visualize larger data sets and embed them into context in a meaningful way. This typically two-dimensional output is accompanied by mouse, keyboard, or touch input. There is usually a high degree of reuse of well-known concepts in new tools to support learning and foster best practices. For example, many tools have overlapping windows, tool bars, context/pop-up menus, save dialogs, or keyboard shortcuts. Depending on the domain, there might also be several unique widgets such as sheet editors in music composition tools (Wright et al. 1997).

Query languages support accessing and preparing data for widgets. They are programming languages with a specific focus. For example, if the information is stored in a relational database, SQL can be used to access those tables and to perform filter and aggregation operations. Given the concept of tables and rows, the expression "SELECT name FROM customers WHERE age > 60" reads the table "customers" and selects the rows with a certain age value to finally return the "name" column. One might easily have a textual or graphical representation in mind when querying data but such languages are independent of presentation. Tools can present the same information in different ways.

Mapping languages are required because the data providers and the interactive widgets usually do not speak a shared language, meaning that there is no inherent graphical representation for many software artifacts. Of course, it is easy to map information to textual representations because many data providers, internally, talk text-only. In information technology, textual representation of information is very important and is standardized, for example in the Unicode standard. This standardization is necessary for the sake of sharing, persistence, and long-lasting comprehension. However, in programmers' minds, some software artifacts have a more vivid appearance than others. Computer graphics employs lines, shapes, colors, or animation to make digital information almost tangible on screen. List widgets, for example, may benefit from "icons" or "color" but the data is only text. Here, mapping languages can be used to "materialize" concepts, that is, for example, mapping the string "red" to actual color information to be displayed on screen.

2.3 Live Programming Environments

A central issue in program comprehension is how the rather abstract source code is put into action when it is executed on the computer. That means, understanding the correspondence between observable program output and its sources is paramount. To achieve this understanding, programmers execute smaller portions of a program called *tests* or pause program execution at certain points referring to (conditional) locations in source code, called *breakpoints*. Assumptions can be verified by exploring a program's run-time state or the result of a test run. Having such a unit of observation, small changes to the source code can also be used to check behavioral variations.

Edit-compile-run cycles can be minimized in live programming environments such as Squeak/Smalltalk. Programmers can evaluate any piece of source code in a text field within a particular context. There is always the global context, which means that **3 + 4** will evaluate to **7** and **Morph new openInHand** will create a blue rectangle attached to the mouse cursor as depicted in Fig. 3. More specific contexts occur, for example, if an algorithm is in the middle of execution and it is paused by breakpoint. In that context, the keyword **self** evaluates to the object holding the shared state the algorithm is working with. For graphical objects, the expression **self color** will then evaluate to the object's current color. Although traditional environments such as Java/Eclipse[5] or C#/Visual Studio[6] do provide context when debugging, Smalltalk environments provide many more opportunities for programmers to work with run-time information. Setting breakpoints is then not always the first choice. Depending on the scenario, it can actually feel like "debugging mode is the only mode"[7], which is not feasible when writing, for example, a Java program.

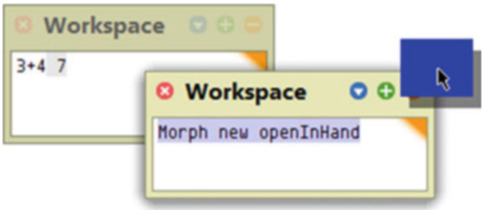


Fig. 3 In Squeak/Smalltalk code can be evaluated in any text field. Here, two workspaces illustrate this concept. The *blue rectangle* is a morph and the result of the execution of the lower code snippet

[5]The Eclipse programming environment, http://www.eclipse.org, accessed on Dec 3, 2015.

[6]Microsoft Visual Studio programming environment, http://www.visualstudio.com, accessed on Dec 3, 2015.

[7]"Debug Mode is the Only Mode", blog post from Gilad Brancha, http://gbracha.blogspot.de/2012/11/debug-mode-is-only-mode.html, accessed on Dec 3, 2015.

Moreover, the Smalltalk programming language is also quite convenient to use as a query language for tool customization. The information is accessed in terms of Smalltalk objects such as morphs or colors as mentioned above. Program execution is structured with objects for processes, classes, methods, class instances, and method activations. Smalltalk can be used to query this information and prepare it for tools.

Smalltalk can be used as a mapping language, too. In Squeak, there is the interactive, graphical system called Morphic. All graphical objects are called *morphs*, which are basically rectangular areas that support composition in terms of holding sub-morphs. Although the boundaries blend, both the Smalltalk language and the Morphic system are important for mapping data to graphical representations. Programmers can describe custom morphs to display any kind of data. The Squeak environment provides tools that support code browsing, writing, and running activities with the help of windows, buttons, text fields, and lists. Every software artifact is an object, every graphical thing is a morph. This is a quite simple yet powerful concept that programmers can employ to accommodate any programming challenge.

The conceptual distance (Hutchins et al. 1985) between *morphs* and interactive *widgets,* such as buttons, is rather long., There are *tool building frameworks* that build on top of Morphic to minimize the amount of source code that has to be written for tools. We will explore our approach Vivide (Taeumel et al. 2014), which is a tool building environment with a data-centric focus, in the remainder of this chapter.

3 Reflect on Your Working Habits

In this section, we describe some prominent, recurrent scenarios where standard tools, which are aligned with programming language concepts, impede the discoverability of tangible examples and hence programming tasks. This should raise programmers' awareness to discover tool building as an opportunity to improve current program comprehension and modification strategies.

3.1 About Finding Tangible Examples

Programmers search for answers to program comprehension questions by navigating, viewing, and collecting software artifacts. Often, the means to navigate, view, or collect presents a challenge. Therefore, programmers should consider improving the situation by adapting the tools involved. A situation becomes challenging whenever programmers have to remember much information, interact with many tools back and forth in a loop, or continually ignore the same redundant or unimportant information over and over again. The screen real estate has to be optimized and necessary user input minimized. Finally, all important information has to be presented on screen so that the programmer can think about the current task with minimal cognitive overhead and come up with a solution involving where to modify the application to fix that bug or add that feature.

Programming tools are the means to navigate, view, collect, and even modify software artifacts. Primarily, programmers have to understand existing source code, modify existing source code, and write new source code. There is also other information that materializes in software artifacts. It originates from the operating system, programming language, execution environment, and other tools. Such artifacts are called files, classes, methods, tickets, emails, traces, processes and so on. They are typically related in one way or another. For example, emails can contain text referring to pieces of source code or traces can contain links to methods from program execution. Such relationships may not be explicit but have to be derived. Tools can help navigate relationships automatically to combine artifacts of different kinds with each other. For example, Mylyn (Kersten and Murphy 2006) achieves this for source code and tasks, represented via tickets in an issue tracker. The tools' interactive widgets can help to display the information in a way that is helpful for the programmer to reveal news or recall what was already known and again of importance for the current task.

There are *facts* and there is *information* derived from those facts based on *rules*. For example, the birthdate of a person is a fact and its current age a derived information. In this respect, the source code comprises many facts. When running code, more data can be derived and interesting properties can be observed. Programmers use tools to learn about facts and also explore derived information. Some rules, however, are implicit and have to be inferred if necessary. For example, the way debuggers acquire access to the current program state is typically hidden in the internals of the debuggers' source code.

Programming environments support intra-tool communication. If the operating system is the programmers' environment, then files are typically used to store source code and exchange related artifacts between editors, compilers, or debuggers. There are programming environments that work on top of the operating system such as Emacs, Eclipse, or Visual Studio. Their means of tool communication enrich the file concept with, for example, text buffers or object-oriented structures. This simplifies the programming model for the tool builder. In Squeak/Smalltalk, programmers are almost completely shielded from the file system and only work in terms of objects, meaning classes, instances, methods, or method activations. While there is still support for text, object-orientation fosters abstract yet domain-specific thinking and also the creation of interactive, graphical programming tools. For example, if the project is about building an address book, then the objects might include persons or addresses and the tools can reflect their relationships with appropriate views and provide appropriate navigation links. A specialized object explorer, for example, might resemble a real-world address book to support program comprehension tasks.

The information space is changing. New source code gets written, and deprecated code gets removed. Yesterday's fact may have become derived from another fact based on some rules. Source code evolves in the way that it gets partially rewritten, called "refactoring." This occurs many times as programmers learn more about the respective problem domain and make changes accordingly. Such additional knowledge about the system or domain is also exchanged via emails or tickets. All kinds of software artifacts are constantly changing, which means that the tools will also have to change to accommodate domain-specific characteristics and personal traits.

3.2 *Search for Examples and Navigate the Results*

In most environments there is a text-based search tool that supports programmers in finding software artifacts by name or (text-based) contents by typing in a search term. Unfortunately, there is usually only a fraction of all information accessible, which is the source code and maybe some external documentation. However, this serves as a valid starting point for many program comprehension or modification tasks (Sillito et al. 2008). Programmers ask questions using terms from the problem domain and expect software artifacts to be named or described likewise in the code or comments. Having such starting points, exploration can continue with, for example, setting breakpoints and running the application.

An important tool for understanding the correspondence between source code and run-time is called "object explorer." In a class-based system such as Squeak, objects are instances of classes that have instance variables for object composition. This composite structure can be navigated because object explorers expose all such instance variables by name and with a textual summary of the particular referred to object, as depicted in Fig. 4. Having this, the programming language dictates the functionality of this tool. Challenges arise when *domain-independent* relationships increase the tool interaction effort or when related artifacts have to be explored in

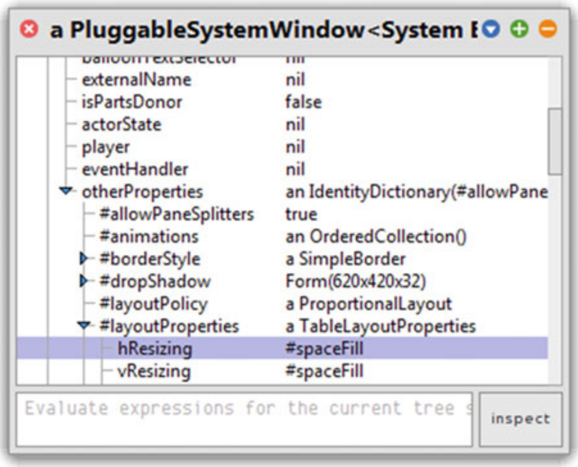


Fig. 4 The object explorer in Squeak. Exploring layout properties of a morph is challenging because morphs hide that state in an extension object and in an additional dictionary structure. In this example, the information is at the third level in the tree structure

separate tools. For example, morphs in Squeak encapsulate several properties, such as related to layouting, in an “extension” structure. Programmers always have to navigate this extension to find out about the current layout. When comparing the state of two different morphs, it is necessary to interact back and forth with two object explorers. However, simple integration points, for example the name of the instance variable, could be used to create a *combined* object explorer.

Refactoring tools require a set of source code artifacts to operate. For example, those tools support renaming or restructuring methods and update all related parts of the code. In dynamically typed programming languages such as Smalltalk, refactoring tasks benefit from additional information, such as run-time types and user-defined filters, to prevent inadvertent code changes. However, embedding a refactoring activity into an exploration activity can be challenging if tool integration is missing. All the different kinds of information involve handling separate tools and hence increase the cognitive effort, and also the risk of making mistakes. Sometimes, a rule that describes integration points can be simple like “Only consider the source code that I’ve modified during the last 2 h.” Such a rule should be manifested in tools to optimize the current program comprehension strategy.

3.3 View Information About Software Artifacts

Programmers perceive software artifacts by viewing a subset of the artifacts’ information on screen. Mainly, there are inherent textual descriptions such as names or numerical values. However, there is usually more information about an artifact available than there is screen space and programmers only want to see the relevant details. Hence, filtering is a common way to customize the tools’ widgets. Many tools anticipate this action directly in the user interface without having to modify their source code.

The *console*, in Squeak called *Transcript*, is a common tool used by programmers for debugging. The practice is sometimes referred to as *printf-debugging* because standard libraries for the C programming language offer the function **printf** to write text to the standard console output. In Squeak, this corresponds to **Transcript show: someObject name**, which, for example, prints the object’s name. Programmers use the Transcript to trace information in the program without having to pause its execution. They have to map data or object structures to text but they can access anything from the particular context. Challenges arise when programmers fail to extract the relevant information and thus have to re-execute the presumably deterministic part of the program. As the output is typically in a text format, there is no other way to explore the underlying software artifacts with this strategy. See Fig. 5 for an example.

There are many tools that require a textual representation of objects for on-screen display such as the object explorer in Fig. 4. When there is a class hierarchy with a common base class—see **Object** in Squeak or Java—programmers can overwrite **#printString** (resp. **toString()**) in a subclass to accommodate

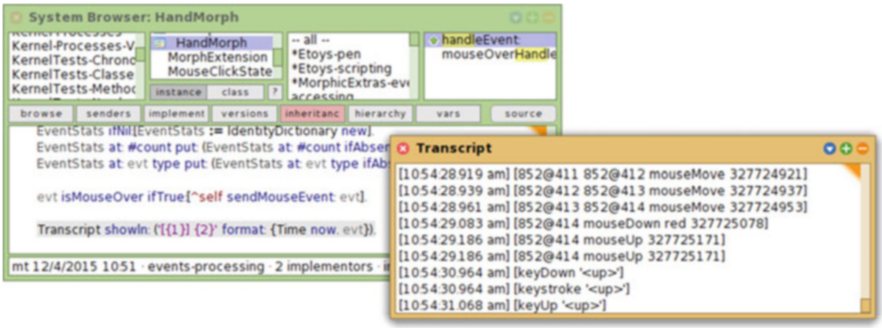


Fig. 5 "printf debugging" in Squeak. If programmers do not extract helpful information, they have to modify debugging statements and re-execute the program

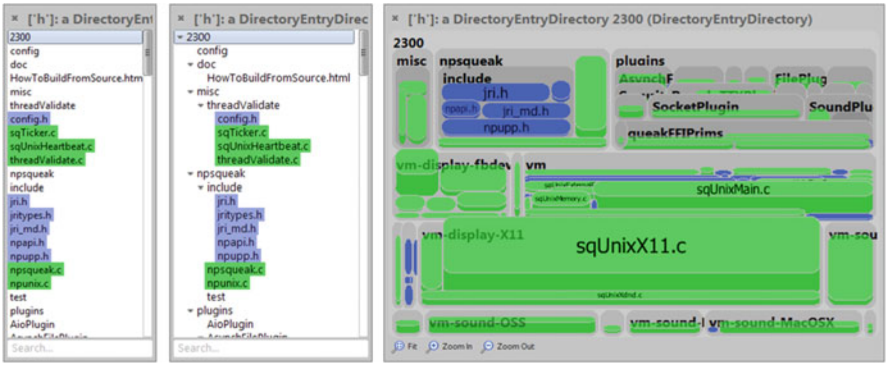


Fig. 6 Three views that display the same software artifacts but reveal different information. The list (*left*), the tree (*middle*), and the tree map (*right*) support labels and colors. The list ignores the hierarchical structure. The tree has much unused whitespace. The tree map has a space-filling approach

domain-specific characteristics. The default textual representation of objects in Squeak is constructed with the class name and the identity hash like "aPerson (1234)," which is rather abstract. In fact, this example is not tangible at all. A slightly better mapping could be "John, Doe, 32." However, the underlying concept of persons would be rather implicit and only discoverable if the programmer associates that information with the concepts of *name* and *age* and eventually with a *person*. A very elaborate version can reveal all details: "aPerson (forename: John, surname: Doe, age: 32)." However, this representation not scale when printed on the console among much other information. It is also independent from any particular programming task or personal preference or existing knowledge. There is a need to adapt the textual representation ad-hoc according to the current situation.

Many programming tools have list-like widgets. Examples include class browsers, search result explorers, and save dialogs. List-like widgets provide an overview and typically some interaction to view, select, move, or drag items.

Although there is often support for filtering, lists usually lack support for shaping an item's graphical appearance, including layout properties. See Fig. 6 for the same artifacts displayed in a list, a tree, and a tree map. Note that it is not feasible to modify **#printString** as described above because programmers may want to see different information in different tools and, most importantly, according to the relevance for the current programming task.

3.4 Collect Useful Pieces of Information

Programmers can take notes in files via text editors, which are part of many programming environments. However, a textual representation of complex object structures requires filtering and summarization, which may be too early to do with the present system knowledge. Programmers may omit to write down important information. This practice suffers from the same problems as printf-debugging, as described above. In Squeak, there is a text-editor-like tool called *Workspace*, which supports source code evaluation (Fig. 3). It also supports dropping graphical objects, which then get captured and are referenceable with a variable like **droppedMorph**. These objects can be explored by evaluating the snippet **droppedMorph explore** within that workspace. It can be beneficial to keep track of interesting software artifacts and defer setting up a representation on screen until later in the course of the programming task when concepts become clearer and examples more tangible.

Tools consist of one or more *windows* to manage their contents. Basically, windows are rectangular areas that show a document or a scrollable portion of it. Tool operations become accessible via push buttons, menu bars, and other additional widgets, typically arranged around a central view. Finding an efficient window manager for programming environments with graphical user interfaces has a long history (Myers 1988). Semi-automatic window layout strategies range from overlapping (Squeak) to tiling (Eclipse) to stacking (Web browsers)—or any combination. Managing the position and extent of tool windows is still part of programmers' frequent activities, and is in the focus of research (Röthlisberger et al. 2009).

While windows can be used to collect and arrange information about software artifacts on screen, their tool-driven characteristics entail much redundant and task-independent information. A window does not correspond to a single software artifact but many. Tool windows in Squeak, for example, are self-contained and display full context information such as code browsers do for displaying *class* definitions and *method* source code. This has the disadvantage that two code browsers that browse two methods from the same class will both display the same context information. This means both will display irrelevant message names, message categories, and other classes in the same package, and class categories with similar spelling (Fig. 7). In Eclipse, tools use a tiling layout strategy and display more specific and less redundant information in their windows such as

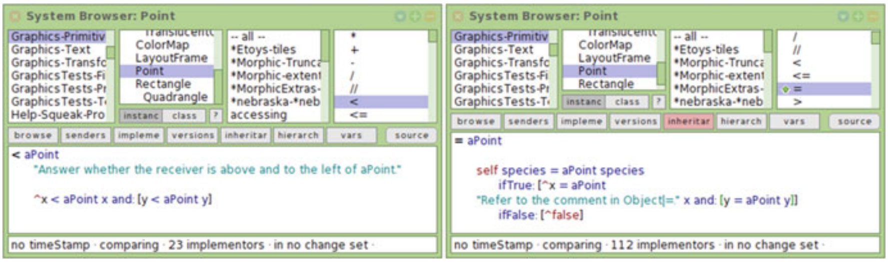


Fig. 7 There is much redundant information visible when comparing, for example, two methods from the same class (here: < and = from **Point**). This is not necessarily a problem of window management but of the tool's user interface design

the class outline at the side and the dedicated text editor in the center. Tiling, however, is much more restrictive when it comes to collecting examples because screen space is limited and quickly exhausted if you can only arrange windows side-by-side. If you stack them, like tabbing in web browsers, you will save screen space but also sacrifice visibility of information completely. Programmers should be able to choose the best way to arrange examples on screen because this level of control can improve tangibility.

3.5 Improve the Means to Navigate, View, and Collect

Tool modification is necessary to accommodate domain-specific tasks and individual system knowledge. A dedicated tool builder who usually creates tools with a *high level of prospective reuse* can only provide means for common tasks, which are usually aligned to the particular programming language and execution environment. Domain-specific characteristics and personal traits cannot be known upfront. Programmer who have concrete tasks, such as fixing bugs and adding features, are best suited to improve working habits by adapting the tools in use themselves.

Thus, whenever navigation repeatedly requires many user interactions, the visual display hides relevant details, or the collection of insights is challenging, programmers can take the chance to act as tool builders. As one would build a level editor for a game to support the creation of game content, programmers can build customized tools to support the work on any software system.

4 Apply Data-Driven Tool Building

Programmers can find tangible examples for comprehension tasks by using programming tools for navigating, viewing, and collecting software artifacts. Such tools are built with a query language to access and process artifacts, a mapping language to cope with inappropriate or missing graphical representations, and a presentation language to serve the user with an interactive front-end. Whenever detecting deficiencies in tools, programmers have the skills to improve those tools ad-hoc—even for scenarios that may be unique.

However, tool building frameworks require a high effort for *tracing* a tool's observable deficiency to the responsible portion of the tool's source code. Additionally, code *changes* are rather verbose and tools do not *update* consistently so that programmers have to restart and hence constantly repeat certain interactions before proceeding with the actual programming task. Eventually, the prospective cost-benefit ratio might render such a tool adaptation pointless and hence programmers keep on using standard tools and inconvenient working habits.

We created a new tool building framework, called Vivide (Taeumel et al. 2014), that projects a data-driven perspective on graphical tools and employs a scriptable way to modify the tools in use with low effort. The extensible presentation language consists of common widgets such as buttons, text boxes, and lists. Both query and mapping language are based on Smalltalk and integrate seamlessly existing object-oriented code. In this section, we will describe Vivide's concepts in detail and apply them for *navigating, viewing, and collecting* software artifacts to increase tangibility of the exemplary information shown on screen.

4.1 The Vivide Tool Building Environment

Vivide (Taeumel et al. 2014) is a programming environment that supports programmers to focus on their domain-specific data. It projects a data-driven perspective on graphical tools and employs scripts that express rules for transforming data and extracting relevant information to be stored in a model. That model will be interfaced from interactive widgets such as lists or buttons. We think that by putting the domain-specific data (resp. software artifacts) in the foreground the notion of tools fades into the background. Programmers are more likely to create or modify tools as an unnoticed side-activity while navigating, viewing, and collecting relevant information.

The Vivide environment provides a direct correspondence between all graphical parts of the user interface and the internal tool logic as depicted in Fig. 8. Vivide is implemented in Squeak/Smalltalk and builds on top of the Morphic framework, which supports direct manipulation of all graphical objects. Every morph has a meta-menu, called *halo*. It can be invoked with a dedicated user interaction such as a click on the middle mouse button, that represents a graphical meta-interface to perform inspection and modification tasks. Vivide makes use of the halo concept to provide access to the underlying tool mechanics as depicted in Fig. 9. Now, programmers can easily find responsible data transformation scripts starting with a visual impression and express modifications in the script source code. Due to this simple yet powerful abstraction, the Vivide framework can update all running tools

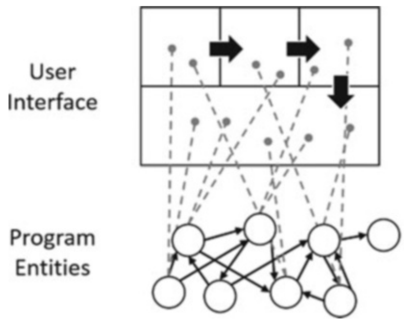


Fig. 8 Traditional tools (*left*) provide a rather complex, indirect correspondence between user interface and program entities; Vivide tools (*right*) provide a direct correspondence. Rectangular portions of the user interface are called *panes*. Programmers have to think about tools being only data-exchanging panes

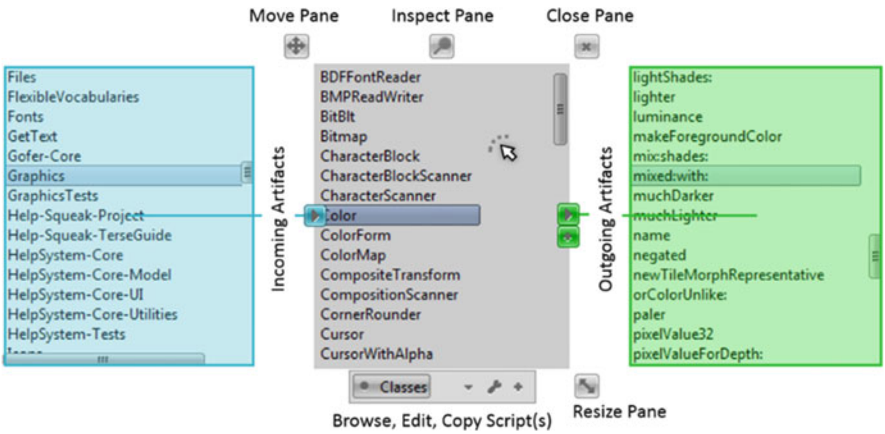


Fig. 9 Every graphical part (pane) of tools in Vivide has a halo that provides access to data-flow properties and the underlying script via floating buttons

consistently when scripts change. The programmer can continue the programming task without having to repeat previous tool interactions.

Thus, Vivide is not only a programming environment but also a tool building framework. Building tools means composing widgets, like in a GUI builder, and writing script code. Script code describes rules for transforming software artifacts and extracting relevant information to be used by widgets. For example, a script that transforms classes into methods and extracts the selector from methods to be displayed in a list can be created with only a few lines of code as depicted in Fig. 10.

There is also a *wizard* involved that tries to detect 1:n, n:1, or n:m transformations to further reduce programming effort. In general, programmers have full control over the input and output of objects in a script. In Smalltalk terms, a script is a collection of *blocks* in the form **[:in :out | "..."]**, and the Vivide framework will

```
{  [:class | class methods]
       -> { #view -> ListView }.
   [:method | { #text -> method selector }].
} openScriptWith: {Morph}.
```

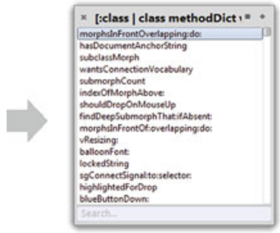


Fig. 10 In Vivide, only a few lines of Smalltalk code (*left*) are needed to describe an interactive tool (*right*). Here the tool operates on the Morph class, transforms it into methods, and displays the methods' selectors in a list that supports drag-and-drop for continuing the exploration

initiate block evaluation with actual objects. The wizard expands **[:a | a + 1]** to **[:in :out | out addAll: (in collect: [:a | a + 1])]** and other expressions to similar constructs.

We think that programmers who build tools with Vivide will benefit from time-related advantages compared to traditional tool building approaches. This may have an impact on the cost-value ratio of tools and thus also on the whole tool building community.

4.2 Support Navigation with Adapted Tree Structure

Scripts in Vivide support describing tree structures to be used by widgets. Each script is a set of object transformation and property extraction rules. With this, programmers can transform any collection of input objects into any other collection of output objects. For example, programmers can navigate *existing relationships* or perform a computation to *derive new information*. The empty script is **[:in :out | out addAll: in]**, which just forwards all objects from the input buffer to the output buffer. After transforming objects, programmers can describe properties of interest such as #text as extracted in the example above. Alternating transformation and extraction means describing multiple levels of a tree structure. While plain list widgets might not take notice of such a tree structure, tree widgets will do as depicted in Fig. 11. Programmers can adapt tree structures to simplify navigation and make efficient use of screen real estate. An example of this is presented in Fig. 12.

Scripts can process multiple data sources as a means of combination and integration. Multiple sources, meaning collections of objects, can be combined as the *Cartesian product*. The scripts still operate on collections of objects but these then contain *n-tuples*, where n depends on the number of data sources. For example, having two sources (1, 2) and (a, b, c), the script then handles [(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)]. Programmers have to know about this in their scripts. The aforementioned script wizard helps combine multiple data sources.

```
{  [:a | a + 5].
   [:a | #text -> a].
   [:b | b even ifTrue: [b / 2]].
   [:b | #text -> b]
} openScriptWith: #(1 2 3 4 5 6).
```

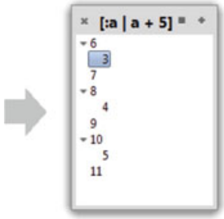


Fig. 11 Alternating transformation and extraction describes different levels (a and b) in the tree structure. Note that Smalltalk code can be used in all scripts. The wizard will expand all four blocks into the form **[:in :out | "..."]**

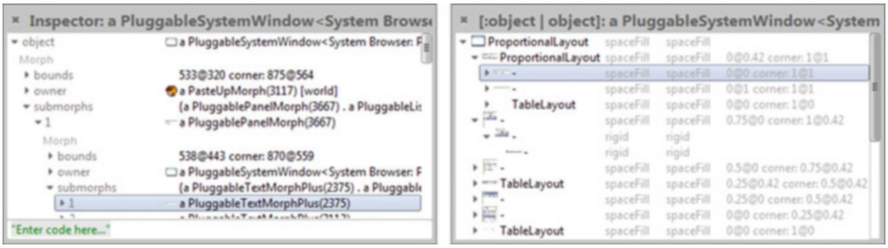


Fig. 12 On the *left* there is a default object explorer showing details about the window of a code browser. On the *right* the same object structure is simplified by only navigating the composite GUI structure (sub-morphs) while exposing layout properties. The same object, a text input field, is selected in both windows to emphasize the efficient use of available screen space

Programmers can then use Vivide scripts to describe arbitrary navigation paths via tree structures. A single tool can shorten navigation paths by only exploiting the relationships of interest as depicted in Fig. 12.

4.3 *Support Views with Named Properties*

Scripts can be used to map any information of a software artifact to something that widgets can use. For example, programmers can derive color information, setup labels or tooltips, and so on. Such *property extractions* have the form of an array with associations as depicted in Fig. 13. Besides visual mappings, any *object-specific* information can be provided for widgets given a name. Common properties include **#text**, **#icon**, **#balloonText**, and **#color**. The widgets decide about those means of configuration and can, theoretically, adapt their whole behavior. It is not part of the concept of Vivide to prescribe the use of such properties. The tree map in Fig. 13 is able to use text, weight, color, and elevation.

Scripts also have a set of *object-independent* properties. For example, Vivide stores a flag **#isProperty** to distinguish between object transformation and property

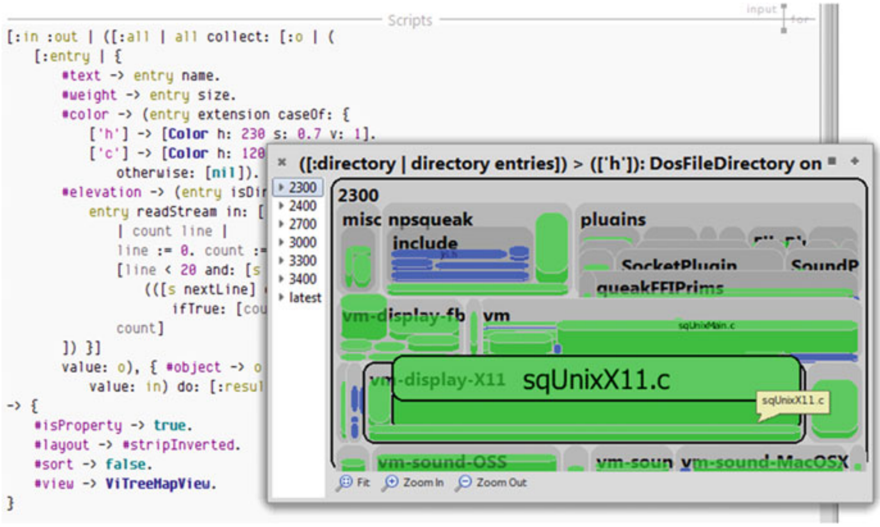


Fig. 13 A rather complex example of a script (*background*) that extracts properties for a tree map (*foreground*). Object-specific properties include text, weight, color, and elevation. View-specific properties include layout strategy and sort order. *Left* of the tree map, the tool also offers a tree representing part of a directory structure with source code files

extraction scripts. Widgets also get the chance to configure themselves according to script-properties. The tree map in Fig. 13 reads **#layout** and **#sort** to adapt its general layout strategy and sort order.

Programmers are in control of the presentation of software artifacts. They can choose between a set of views such as lists, tables, trees, tree maps, and other charts. Furthermore, they can tweak those views at the script level. Any change in script code will immediately update the corresponding views in the programming environment. Such short feedback loops support programmers in exploring information and finding examples. Those examples can become tangible if programmers can find an appropriate way to show them on the screen—tailored to the domain and personal preferences.

4.4 Support Collection with Arbitrary Containers

Vivide supports overlapping windows for tools like Squeak does. However, Vivide entails the idea that "window management" can be pluggable. Based on the authors' experiences, overlapping windows suffice most of the time. Common tool windows, however, have their contents tiled or stacked as depicted in Fig. 13. Within those tiles, the layout strategy may be different. List widgets, for example, arrange their items vertically side-by-side. In another example, tree maps can have

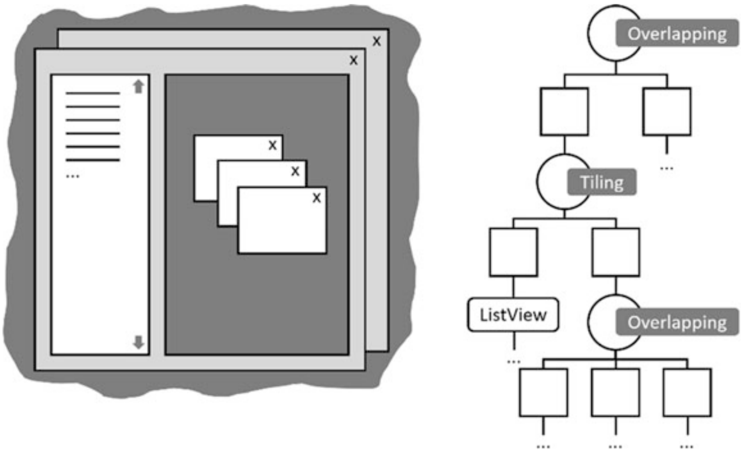


Fig. 14 Window management should be pluggable and any layout strategy applicable in any part of the graphical hierarchy as needed. Here, the hierarchy on the right models the tools on the *left*. In the hierarchy, *circles* denote *multi-pane widgets*, *squares* denote *panes*, and *rounded squares* other *morphs* in the world

overlapping items when supporting elevation—or even a 3D canvas. Based on these observations, we think that programmers should be able to decide about the layout strategy for each level in a tool's graphical hierarchy. The concept of rectangular tool building blocks, called *panes*, in Vivide is expanded to *multi-pane widgets*. These widgets encapsulate multiple panes to apply any possible layout strategy. For an example, see Fig. 14. Such means of content organization are independent of actual visualizations. Programmers can employ them as required to make software artifacts more tangible on screen.

Collecting objects via drag-and-drop with a pointing device (mouse or touch) is a common practice in Vivide. Many widgets support dragging their displayed objects. Then, there are containers that help collect those objects as depicted in Fig. 15. Programmers can collect pieces of source code from across the system and view them side-by-side as a convenient, problem-centric representation. Programmers can also mix different kinds of objects such as run-time artifacts and code artifacts. Although Vivide provides support at the level of graphical, interactive tooling, the Squeak/Smalltalk environment is the reason for run-time information being omnipresent in general.

5 Conclusion

Program comprehension benefits from tangible examples. Due to the abstract characteristics of source code, tangibility does not refer to a physical representation but to conceptual clarity and understanding. Regarding a concept in the source

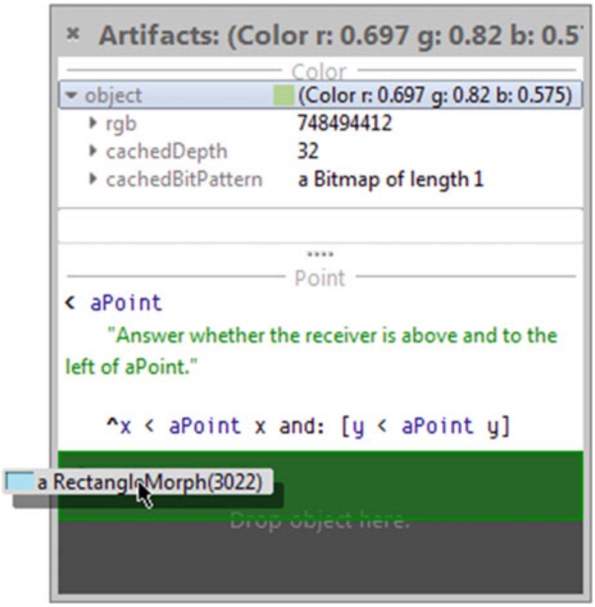


Fig. 15 Vivide provides a generic container to collect software artifacts via drag-and-drop. Each dropped artifact is displayed in an interactive view. Here, there is an object explorer for a color object, a code editor for a Point method, and a mouse cursor about to drop a rectangle morph

code, its correspondence in the problem domain has to be understood as well as its effects during program execution. Having this, finding the subset of relevant information and presenting them in a tangible way is a matter of efficient programming tools and programming environments. This efficiency is specific to the domain and to the programmer's personal preferences and existing knowledge. Traditional programming tools, however, align with generic programming language concepts; specific scenarios are not well supported. Tool adaptation seems beneficial but is typically expensive.

We presented the Vivide programming and tool building environment, which is a data-driven, scriptable, interactive approach to construct graphical tools for programming. We applied Vivide in several examples to illustrate ways to improve the means to navigate, view, and collect software artifacts. While programmers do not have to come up with the perfect solution right from the beginning, Vivide's direct feedback after each tool modification fosters an iterative and explorative working mode. Programmers can safely try out ideas and undo mistakes with ease. Even unique scenarios can be improved, and reuse can then become of secondary interest.

References

Cohn M (2004) User stories applied. Pearson Education Inc., Boston, MA

Courage C, Baxter K (2005) Understanding your users: a practical guide to user requirements. Elsevier, San Francisco

Csikszentmihalyi M (2008) Flow: the psychology of optimal experience. Harper Perennial Modern Classics, New York

Hutchins EL, Hollan JD, Norman DA (1985) Direct manipulation interfaces. Hum-Comput Interact 1(4):311–338

Kersten M, Murphy GC (2006) Using task context to improve programmer productivity. In: Proceedings of the 14th international symposium on foundations of software engineering (FSE)

Myers BA (1988) A taxonomy of window manager user interfaces. IEEE Comput Graph Appl 8 (5):65–84

Norman DA (2002) The design of everyday things. Basic Books, New York

Röthlisberger D, Nierstrasz O, Ducasse S (2009) Autumn leaves: curing the window plague in IDEs. In: Proceedings of the 16th working conference on reverse engineering

Shneiderman B, Plaisant C (2010) Designing the user interfaces: strategies for effective human-computer interaction. Addison-Wesley, Reading, MA

Sillito J, Murphy GC, De Volder K (2008) Asking and answering questions during a programming change task. IEEE Trans Softw Eng 34(4):434–451

Taeumel M, Perscheid M, Steinert B, Lincke J, Hirschfeld R (2014) Interleaving of modification and use in data-driven tool development. In: Proceedings of the 2014 ACM international symposium on new ideas, new paradigms, and reflections on programming & software

Von Mayrhauser A, Vans AM (1995) Program comprehension during software maintenance and evolution. IEEE Comput 28(8):44–55

Weiser M (1987) Source code. IEEE Comput 20(11):66–73

Wright J, Oppenheim D, Jameson D, Pazel D, Fuhrer R (1997) CyberBand: a ‘Hands On’ music composition program. In: Proceedings of the international computer music conference

Case Studies on End-User Engagement and Prototyping during Software Development

An Overview of Current Practices in the IT Industry

Franziska Dobrigkeit, Sebastian Meyer, and Matthias Uflacker

Abstract Today software reaches into almost every aspect of our lives, with mobile devices and their apps a major source of our everyday software experience. This trend has shifted our expectations toward a more user-friendly, intuitive and easy to use user experience. Thus appealing user interfaces and an excellent usability become key to successful software products and services.

However, great usability and user experience are not easy to develop, because system engineers traditionally design solutions without involving end users. On the other hand, current research suggests the involvement of end users in software development and the constant incorporation of testing and feedback to provide high-quality software and satisfying usability. Based on these facts, and with a rising awareness for user experience, companies are incorporating user-centered approaches, such as Design Thinking, which involve end users and other stakeholders in the development process. Most of these user-centered approaches are strongly based on early research and validation with end users and prototypes with different levels of fidelity.

This chapter provides a look into the development process of three major software companies and presents an overview of their current practices concerning end-user involvement and prototyping.

1 Introduction

Good usability and user experience are not easy to develop because systems engineers often design solutions without involving end users (Riley and McConkie 1989; Nielsen 1993). As a result, solutions are technically feasible but do not necessarily meet the needs and requirements of users. This phenomenon has often

F. Dobrigkeit (✉) • S. Meyer • M. Uflacker
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: franziska.dobrigkeit@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_12

been proven in research studies (Budde et al. 1992; Rasmusson 2010; Norman and Draper 1986). Current research suggests the involvement of end users in software development to provide high-quality software and satisfying usability (Hornbaek 2006; Rasmusson 2010; Coursaris and Kim 2011). Besides highlighting the importance of end user involvement, current research also suggests the constant incorporation of testing and feedback (Hornbaek 2006).

Since the awareness of good user experience is increasing, companies have been incorporating user-centered approaches such as Design Thinking. These approaches involve end users and other stakeholders in the development process. Most of these user-centered approaches are strongly based on early research and validation with end users and prototypes with different levels of fidelity. Hereby, the creation of prototypes allows for necessary testing before the products are fully developed (Budde et al. 1992; Riley and McConkie 1989). Additionally, uncovering conceptual errors in a product before it has been developed saves resources and money (Nielsen and Budiu 2013).

Over time, researchers have developed different approaches and tools to support user testing. Some testing tools, research methods, and test settings are well researched and widely used, but so far there has been no standard approach of involving end users or using prototypes throughout software development processes. This chapter will give an overview of the current state of end user involvement and prototyping in the software development process. To this end three case studies with major software companies are presented and discussed.

In Sect. 2 the research approach is described and in Sects. 3, 4, and 5 the case studies are presented. A discussion and evaluation of the case studies is provided in Sect. 6. Section 7 closes the chapter with a summary.

2 Research Methodology

Several research methods have been employed to investigate how companies use prototypes and incorporate end users in their software development processes. Companies were initially selected based on the following criteria:

- Companies needed to have at least 500 employees in order to be considered a large company (Thomas 2012).
- It was necessary that several hundred employees be involved in software development.

Based on these criteria three case companies were found:

1. SAP SE is the third largest software company in the world. SAP's core business is standard business software, which is rapidly entering the mobile market.
2. Microsoft Corp. is the world's biggest software company and offers a variety of software products for consumers and enterprises. Moreover, Microsoft develops its own operating system for mobile devices.

3. Nokia HERE is highly experienced with products for mobile devices since its core business focuses on maps and location-based services.

As all companies employ thousands of people, it was important to get into dialogue with people relevant to the focus of prototyping and end-user engagement. Thus, the roles identified as interesting in the software development process were: user researchers, and user experience designers, or similar positions where activities were performed that engaged the end user.

Based on these criteria a total of 25 qualitative interviews were conducted with relevant employees of the three case companies. Several onsite visits were also undertaken as well as discussions with other company employees.

2.1 Qualitative Interviews

The main research method used for data collection was qualitative interviews. These interviews were conducted according to a semi-structured interview guide to allow, first, a comparison between the case companies and, second, to ensure that similar topics are covered in each session. If possible, interviews were conducted in person at the workplace to understand the context and also to make observations. However, due to huge distances, some interviews were conducted via phone or conference tools.

2.2 Observations

In the case of face-to-face interviews, the results could sometimes be enriched by visual impressions and observations made on site. Gaining insights through the actions and behavior of people, rather than relying solely on the statements of interviewees, is a meaningful technique.

2.3 Data Evaluation and Topic Extraction

Most of the interviews could be recorded with an audio device and were transcribed afterwards. For interviews without a recording, handwritten notes were used as reference for further data evaluation. To ensure traceability of the individual information, an anonymous reference was added to each statement, which was kept throughout the entire evaluation process.

The statements for each case study were clustered individually according to topics from the interview guide. The final topics and content blocks for the case studies were then derived from these clusters.

3 SAP SE

SAP is the third largest software company in the world and the leading provider of business software and mobile business applications. Established in 1972, SAP had about 66,500 employees worldwide in 2014. About 17,500 SAP employees have a position in research and development. SAP became famous for business applications for large enterprises in the context of the *SAP Business Suite*[1]. SAP has since expanded its portfolio enormously. Nowadays SAP offers a wide range of standard software for cloud, analytics, and mobile, which is based on its database technology HANA[2].

The organizational structure of SAP has become increasingly complex as the company has grown (Steiner 2012, p. 48). Figure 1 depicts a simplified organizational structure of SAP, including all departments involved in this case study as well

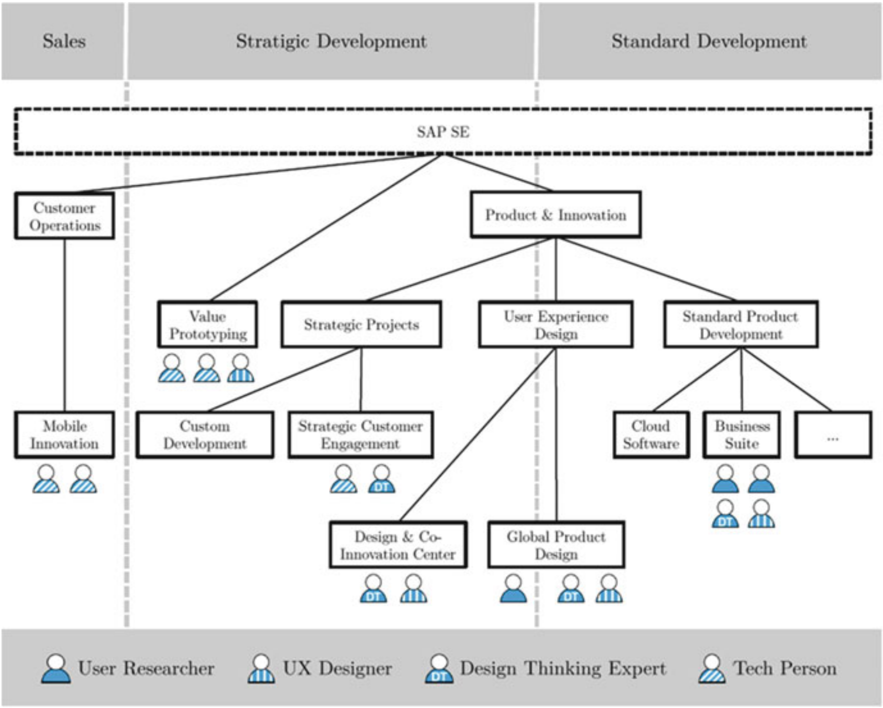


Fig. 1 Simplified company structure of SAP based on the affiliation of interview partners involved in this case study

[1]https://www.sap.com/germany/solution/lob/finance/software/business-suite-apps/

[2]http://www.saphana.com/

as the interview partners and their roles. The term UX Designer summarizes designer roles and the term Tech Person summarizes development roles.

As visualized in the diagram, all product development for standard software can be found in the branch *Product & Innovatio*n. The different product lines, such as *Business Suite*, are subordinate to the *Product & Innovation* department. This branch will be summarized as standard development throughout this case study. In fact, SAP does more than 80 % of its sales and revenue with standard software. Figure 2 depicts a simplified structure of the *Business Suite* department as an example of standard development. The department implements two horizontal teams, the overall suite product team, and a user experience team. Product development is divided according to the line of business categorization, wherein several development teams are grouped.

In cases where customers miss a certain functionality in standard software they can have it developed by *Custom Development*, another branch of SAP. *Custom Development* customizes standard solutions for individual customers or implements solutions for completely new use cases. Several smaller departments such as *Value Prototyping*, *Design & Co-Innovation Center* or *Strategic Customer Engagement* offer services to increase customer involvement in raising sales and providing higher quality software based on customer feedback. *Custom Development* and these different departments are summarized as strategic development throughout this case study (see Fig. 1).

3.1 Software Design and Development Roles

At SAP there are a variety of roles defined for product development with different skills and responsibilities. In the following, only roles relevant to prototyping and end user integration are described. Development teams follow an agile Scrum approach and principals of Lean Development which implement typical roles such as Scrum Master and Product Owner. In addition, there are global product owners, who are responsible for the strategy and the success of the product. Furthermore, the product teams contain quality managers who are responsible for the software quality of several development teams. For the actual development there are several roles defined with the responsibility of developing productive code and tests. Interviewees remarked that there is a trend toward integrating developers into prototyping teams to enable them to build more interactive prototypes.

In addition to development roles, roles for design and user involvement are implemented. These include user researchers, visual designers, interaction designers and user experience designers. Such roles often support development teams in a horizontal organization, but some departments have started moving them into product teams. Finally, the role of Design Thinking expert represents an important area of end user involvement and product innovation. These experts help product teams with their expertise and facilitate workshops with customers to find right use cases or features for existing or new products.

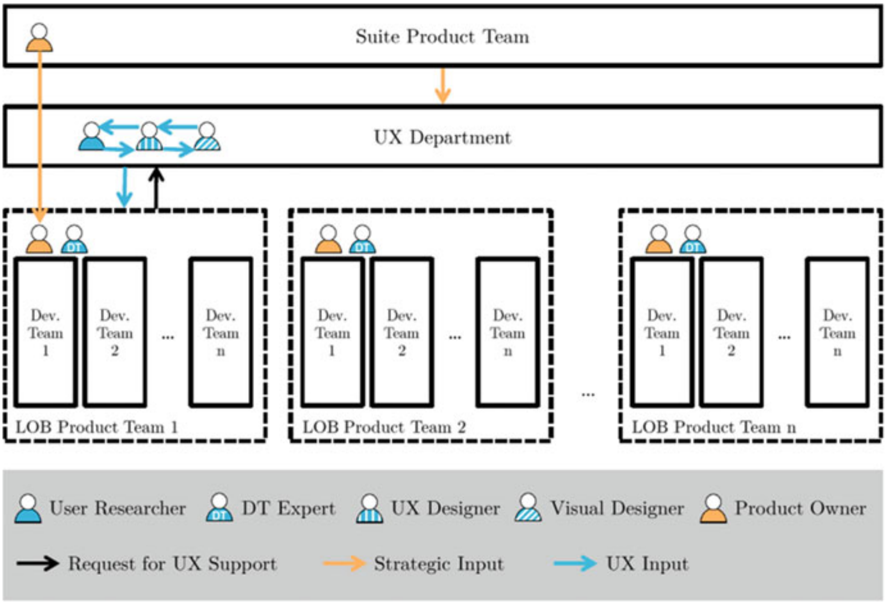


Fig. 2 SAP Business Suite structure as an example for standard product development

3.2 Development Process

The processes for standard development and strategic development are different. In general, it is easier to adopt approaches for strategic development because—in comparison to highly standardized software—there are less policies and guidelines that apply. Since 2009, the default approach for development teams is the agile Scrum framework complemented by Lean values (Steiner 2012, p. 48).

3.2.1 Process for Strategic Development

As different departments at SAP involve customers in a co-innovation mode, the variations of processes and the methods used are very diverse. However, most co-innovation projects follow a similar structure and are often conducted in a short time span of only a few weeks.

1. Initial contact to discuss the project and define the scope is established and followed by a Design Thinking kickoff workshop.
2. Project teams at SAP involve experts as researchers, designers, product specialists to create an early prototype and a customer business case.
3. Concept prototypes are developed iteratively with customer feedback sessions in between or in a co-creation with the customer.

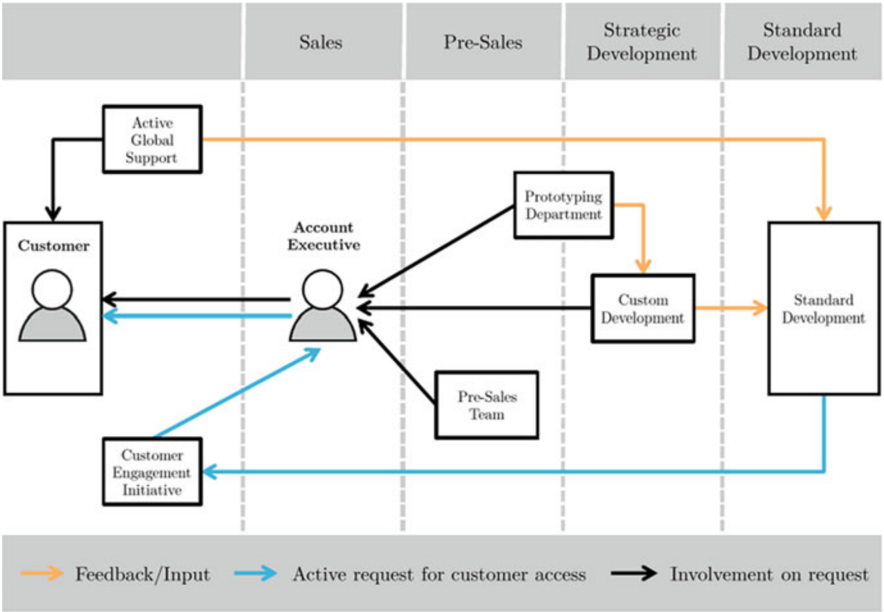


Fig. 3 Customer input and feedback channels for development at SAP

4. Once the customer has agreed on the concept and visual designs, a proof of concept implementation (POC) is developed by one of SAP's prototyping departments or as a co-development project with the customer.
5. More often these customer-specific proof of concept implementations are used to show customers potential business value and sell a customized development project. In such cases, the POC serves as part of the specification for the *Custom Development* department.
6. The result is analyzed and a business case for SAP is created. If the business case is promising, SAP starts to adopt the use case, transferring the learnings into standard development.

Customer integration for this process is channeled through the account executive or Active Global Support (see Fig. 3). The development process of the custom development projects is much more specified. It has to follow additional guidelines and needs to include support and maintenance. The process is not described in further detail as the elements of stakeholder and end user involvement closely resemble the process for *Standard Development*.

3.2.2 Process for Standard Development

The process for standard software development is more structured and more uniform and it has to conform with various guidelines and policies (e.g. internationalization). The following describes an abstracted version of the standard process for the *Business Suite* organization.

1. *Business Suite* management decides on the product portfolio for the next release cycle.
2. Design Thinking activities are employed to create business cases and a backlog in the form of user stories.
3. Software development is started with an agile process and ideally uses best practices such as test-driven development, continuous integration, etc.

This process is supported by a dedicated team which provides tools and services to support the development in all phases. Theoretically, the development process frequently involves customers. In reality it is challenging for development teams to engage with customers and end users because access to customers is restricted and channeled. Figure 3 shows these channels. *Standard Development* is required to actively request access to the customer for feedback activities through the centralized *Customer Engagement Initiative*. This initiative provides contact to relevant candidates for user testing and provides budget and expert support. However, only customers that have signed a cooperation and feedback agreement with SAP can be approached for feedback or testing of any kind by *Standard Development*. Additionally, *Standard Development* gets input and feedback about products from the departments *Strategic Development* and *Active Global Support*. Internal processes for standard development are constantly evolving to be more user-centered.

3.3 Prototypes at SAP SE

The term prototype has a wide range of meanings within SAP and is discussed to a great extent internally because its understanding in the various teams is so different. It can be observed that the understanding of this term is related to the role of a person and his or her department.

Design Thinking experts use the term prototype for everything that can be created within a few hours, for example, paper sketches, tangible prototypes and everything with a higher fidelity. A prototype comes with no guarantee that it will function correctly. People with a background in user research have a similar understanding of prototypes and use the term to summarize artifacts from paper to proof of concept implementation. Interaction and visual designers of the Business Suite, are mainly involved with interactive prototypes (e.g., clickable PowerPoint, HTML or implemented prototypes) and define a prototype as something that is not a finished product.

In contrast to the above, there are departments, such as Value Prototyping, which specialize in building technical prototypes in a short time span of a few weeks. Here the term prototype is used for running software such as technical implementations of a concept or use case. In such departments wireframes or mockups are not seen as a prototype. These artifacts are seen as milestones on the way towards the final outcome: a technical prototypical implementation on the final technology stack. Showcase or proof-of-concept implementation are used as synonyms for the term prototype. However, even within these departments, depending on their backgrounds, designers consider wireframes and mockups as prototypes.

3.3.1 Prototype Usage

Throughout the development lifecycle there are various kinds of prototypes used. For early phases, design thinkers and designers rely on low-fidelity prototypes, because prototypes with a fidelity that is too high can trigger the wrong kind of feedback. During these phases, prototypes are used, for example, for use case definition and as a tool to gather requirements with customers or to internally communicate early concept ideas. Prototypes with a medium fidelity, such as mockups or visual designs, are used by prototyping departments to validate concepts with stakeholders. As the used tool and technology stacks do not matter for these purposes, people use what fits their skills and the current project best, including paper or tools such as Axure or Pidoco, clickable PowerPoints, or HTML prototypes.

Most technical prototypes developed in co-innovation with the customers aim to make SAP technology more attractive and provide a better user experience. Overall, the goal is to increase sales of existing standard software, which is often empowered by a prototype, or to sell custom development projects. These prototypes are used during the sales process: either directly with a customer or as a success story for other customers. In the case of a subsequent custom development project, the prototypes are used as one part of the specification. Another reason for technical prototypes is to show and prove the technical feasibility of a new use case, which is also intended to convince a customer of the salability of the corresponding product.

Since the timespan for projects with customer engagement is very low, the resulting prototype is not intended to be used productively. Therefore, neither wireframes nor the final prototypes are tested during a classical usability test. However, the prototype and early forms of it are used to gather feedback from stakeholders. Such technical prototypes are often based on the final technology stack and other SAP products and, ideally, also use real data.

3.4 Research Methodologies

Quantitative data and the results of general research studies from those others than the interviewees are used to make decisions about the product portfolio. Classical research for standard software products is generally conducted with functional prototypes and based on scenarios for different use cases. This is typically lab-based research following a test script, consisting of an internal test run and a number of subsequent user tests. The results are analyzed and then handed over to the development team. Moreover, some field research is done and customers are involved in workshops to gather requirements and work in co-creation with SAP. Interviewees stated, that participants are often asked to think out loud to better understand their perception. The usage of the "think aloud method" does not depend on the stage of the project and is used for all kinds of prototype testing. Additionally, some analytical tools are in place for web-based products to collect as much data as possible before and after the release of a product. These tools enable the product team to see which features are actually being used.

3.4.1 Workshops with Stakeholders

Although, workshops are not a research methodology by definition, they can be seen as a research method in a broader sense because they are used to gather requirements and insights from users and other stakeholders. At SAP there are different workshop settings with customer involvement. Prototyping and customer engagement departments conduct co-innovation projects together with customers to develop a proof of concept for a new business case. At the beginning of such a project one or more design thinking workshops are conducted with stakeholders and, if possible, end users, with the goal to generate requirements and an early conceptual prototype. Sales and presales departments conduct Design Thinking workshops with customers to give potential value and generate sales opportunities without them being perceived as sales by the customer. Design Thinking workshops for the product development teams are also conducted with stakeholders to find new use cases and learn about user behavior. A smaller form of workshop or interactive feedback session with customers is conducted based on mockups or wireframes. Such sessions consist of a prototype presentation by SAP followed by discussions with important stakeholders and sometimes end users. Rapid prototyping with end users has been mentioned by researchers of the product development departments as another form of user involvement in a workshop setting.

3.4.2 Lab-Based Research

There are two different types of usability tests that resemble lab-based research. In most cases the user is given exact instructions for small tasks within a certain scenario and asked to complete the task. The participant is observed while performing the tasks and if he or she gets stuck or does not understand the prototype the test leader provides help. Usually these kind of tests follow a test script to ensure comparability. In another type of usability test the user is confronted with an app and given no further guidance or instructions. The purpose of this free form test is to see if the user can find and understand the application without training. Both tests are combined with questionnaires before and after the test as well as structured interviews.

Own remarks	Remarks of the test person	Remarks of the test leader
Remarks of the minute taker	Stated expectation	Asked questions
Observations on the test person	Remarks and questions	Possible support
Points of interest		

Fig. 4 Typical document for note taking during qualitative interviews

Interviewees of prototyping departments stated that real lab-based test are rarely conducted due to the effort in terms of time and budget. Instead, the participant is seated in front of a test computer and a small webcam placed next to the participant. The researcher is seated next to the participant to allow for easy conversation. This setup allows for tests to be done in any neutral environment, for instance in a conference room, a booth at a customer fair or during a user group event. For these kind of tests typical data sources, such as video, audio and interaction on the screen, are captured. Usually, the software Morae[3] is used to record the video and screen simultaneously. Although, Morae provides a lot of additional functionality, for example for annotations, only the basic set of features is used during the tests.

There is always at least one other person attending the test to take notes about comments and user reactions. The note taker makes notes either in handwritten form or directly on a computer, the latter being preferred since it allows easier processing afterwards. A typical excel sheet for note taking is shown in Fig. 4. Moreover, tasks are categorized according to how well the task was solved. In order to quantify this information each task is mapped on a scale of 1–4, whereby 1–3 represents the level of help and 4 indicates a complete failure. Interviewees found this data source to be the most important.

3.4.3 Field Research

There are two scenarios for field research at SAP. In the first case, user researchers conduct ethnographic studies in the field to understand users and to define target groups, use cases and business cases. Their processes and use of tools are designed to support their work. These studies are conducted without a specific product or prototype and are mainly based on observations at the customer site. If researchers are shadowing people, they ask questions for clarification or to understand why a user got stuck. The second but rarer scenario is to conduct usability tests in the field. This field research involves much higher effort in terms of both time and money but is preferred by researchers.

[3]https://www.techsmith.com/morae.html

3.4.4 Remote Testing

While remote sessions with customers are a common tool to collect feedback, they are, according to the interviewees, the least preferred way to involve end users. This is because of the huge lack of feedback. However, due to globally distributed teams and an insufficient time and travel budget, remote sessions are often the only way to engage with end users. Especially early stage prototypes, such as wireframes or visual designs are validated remotely. This allows the continuing involvement of the same people. During these sessions the prototype and its concepts are presented to the customer, followed by discussions and questions from the customer.

Sometimes usability tests are conducted remotely by giving the participant control of the prototype via a conference tool. Such tests are hard, as they lack a lot of nonverbal feedback and suffer from technical limitations (e.g. connection quality). Such sessions are often recorded for later replay with the presentation tool SAP Connect.

Another form of remote testing is to share technical prototypes with stakeholders, that run and can be used without assistance. Their feedback is received in textual form and sometimes with additional screenshots via email. Interviewees pointed out that it would be beneficial to know what the users have actually done in comparison to the feedback they provided.

3.4.5 Test Participants

Researchers conduct ethnological studies with end users to provide a basis for use cases and business scenarios, which provide the basis for many products. The results of these studies are then often used for internal Design Thinking workshops to define the portfolio.

Co-innovation workshops are usually conducted with important stakeholders on the customer site, but teams advise customers to also involve end users in these workshops. According to some interviewees, it is much easier to involve end users for feedback at this stage than it is later in the process. This is the reason end users are sometimes involved in co-innovation workshops. Interviewees stated that due to budget and time constraints it is not possible to sell and justify early stage research activities to customers. In the further project progress, milestones such as wireframes and visual designs are usually only validated with the original stakeholders but rarely with end users. Furthermore, the final acceptance test is more on a functional than on a conceptual level. This is because it is done with the same stakeholders who were involved during the entire process. To uncover conception and usability issues for proof of concept implementations, internal design reviews are conducted with colleagues who are not directly involved in the project team but can give valuable feedback based on their expertise and experience.

For standard software products, usability tests and feedback sessions are conducted with both end users and internal colleagues. Researchers within the product teams try to conduct usability tests with end users. As described in Sect. 3.2.2 customer contact is managed by the account executive and product teams, who need to request contact through the customer engagement initiative. As end user tests are costly, valuable internal test runs are conducted to ensure an execution without errors.

4 Microsoft Corporation

The following case study was conducted with Microsoft Corp., the world's biggest software company with more than 100,000 employees. Microsoft operates in a broad range of business areas divided into five large segments, namely: Windows Division, Server and Tools, Online Services division, Microsoft Business division, and Entertainment and Devices division (Microsoft 2013).

The case study is based on four qualitative interviews with employees from different departments who all have the role of Design Researcher or User Experience Researcher and are frequently involved in prototyping activities. Two interviewees are part of the Mobile Team and involved in the development of Microsoft's mobile phone operating system: Windows Phone[4]. The Mobile Team is also responsible for defining the experience of various key mobile applications for the Windows Phone platform. Another employee is part of the Office organization, responsible for the OneDrive PowerPoint Web App as well as PowerPoint on mobile devices. The fourth person is currently working on mobile games and used to work in the Windows Phone and Productivity Applications Team. In addition, to the interviews an onsite visit to the Microsoft Development Center Copenhagen (MDCC)[5]—the biggest development site in Europe—made it possible to conduct an interview with a senior program manager of Dynamics AX.[6] It was also possible to observe and discuss the development activities of one development team.

Microsoft itself has a variety of products and departments for both business and consumer software, which are fundamentally different. Therefore, the insights of this case study cannot be generalized for the whole company.

4.1 Software Design and Development Roles

At Microsoft, there are three main software development roles implemented in the organizational structure: Program Manager (PM), Software Development Engineer (SDE) and Software Development Engineer in Test (SDET).

[4]http://www.windowsphone.com

[5]http://www.microsoft.com/da-dk/mdcc/default.aspx

[6]http://www.microsoft.com/en-us/dynamics/

The most versatile role is the Program Manager, who is mainly accountable for the success of the product and the team. PM's are responsible for setting the focus and scope for a new feature or product, as well as managing internal resources and ensuring the delivery of the product. SDEs are responsible for writing code and developing algorithms for products but are also involved in building technical prototypes. Other than SDE's, SDETs are not involved in writing productive code but ensure code quality by developing tests, internal tools and ensuring the integration into the ecosystem. In addition to the three software development roles, there are two more technical roles namely User Experience Researcher (UX Researcher) and User Experience Designer (UX Designer). User Experience Designers create the interaction and visuals for features or products from very low-fidelity to high-fidelity prototypes and the final visual designs. This iterative process happens in close collaboration with UX Researchers, who conduct ethnographic research, usability studies and the validation of concepts with the help of prototypes. They choose the research methods and create hypotheses but do not usually build prototypes.

Both designers and researchers are embedded in the product teams but report to an overall research organization (see Fig. 5). In addition, they are part of a community of user experience people who exchange best practices. Since they are integrated in the product team, they collaborate closely with SDETs, SDEs and PMs. Other research teams dedicated to conducting psychological and anthropological studies as well as market research, are set up globally in the company. These

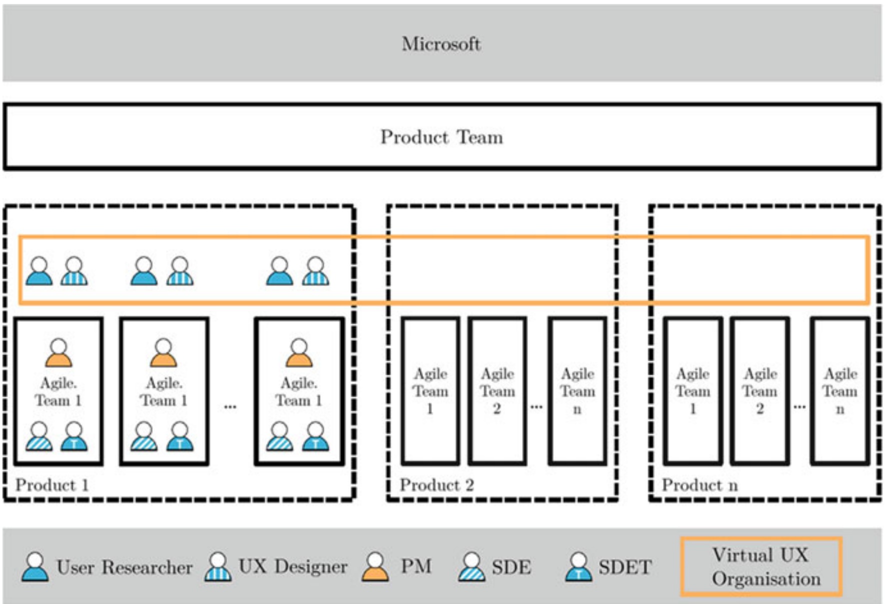


Fig. 5 Software development role integration into the organizational structure at Microsoft

teams provide fundamental research for product segments and are assigned to a project if necessary.

4.2 Development Process

The product development process can be separated into three major phases: planning, design and, execution. In each phase the program manager plays an important role in coordinating resources and pushing first ideas to a product release.

1. Planning Phase: This phase aims at creating a frame in which the future product is built. The goal is to identify key aspects such as potential customers and their needs. To this end the PM communicates with ethnographic researchers and analyzes the whole environment of a new product with a variety of sources. These include: customer studies and surveys, customer satisfaction indexes, personas, usability data, partner feedback, industry analysis, competitive product reviews, focus group results, and more.
2. Design Phase: The design phase is an iterative process with four sub phases. During the exploration phase, the UX designer, UX researcher and PM will create a lot of new ideas to address a particular problem through ideation and brainstorming sessions. The resulting prototypes usually have a low fidelity with the focus on discovering and evaluating potential solutions. In the following phase the discovered design directions are evolved into early concepts. The fidelity of the prototypes increases through iterative ideations and prototype testing. In the definition phase, the concepts are narrowed down to one or two possible solutions. Finally, one design is refined to the desired experience in the solution phase.
3. Execution and Implementation Phase: Once there is a final design proposal for a problem, SDEs get involved. Additionally, SDEs can be involved in earlier stages to ensure the technical feasibility of an evolutionary prototype. Traditionally, this phase is structured similar to the waterfall model. In an initial coding phase, the specifications of the design phase are built. At some point, the product is called "code complete." In the following stabilization phase the software is tested and bugs are fixed. The resulting public beta is followed by another coding and another stabilization phase. Finally, there is a quality assurance phase before the release of the product.

Throughout the phases described above, the UX researcher and UX designer are involved to support the product teams with end-user-engaging activities. They work across different phases of the software development lifecycle because they support several projects within the organization in parallel.

4.2.1 Transition to Agile Development

The product lifecycle previously described in Sect. 4.2 has been implemented for many years, especially for traditional products such as Windows, Office and Visual Studio. However, Microsoft is currently undergoing a significant change in direction to transition the company into a more agile and nimble organization. In this way, Microsoft will better be able to react to fast changing markets, technology trends and also to gain a more competitive edge. One of the most important aspects of its competitiveness lies in its ability to provide appealing user experiences throughout the product portfolio. Therefore, the release cycles have to be much shorter and customer involvement needs to be integrated throughout the whole process (Bright 2014). The change towards an agile work mode can already be observed in some divisions (e.g. in the merging of different software development roles into one software engineering role). Starting as an experiment in the Bing[7] division, SDEs and SDETs have been consolidated to a combined software engineering role. This role has also been successfully implemented in other divisions and can be expected to be adopted in further teams as well. The elimination of the SDET role does not imply that the software is not tested anymore. Software engineers test their own code while quality assurance becomes more important and moves from programmatic automated testing towards end user validation in a real world context (Bright 2014).

As announced by the CEO of Microsoft in July 2014[8], there will be further change to streamline the development process. This change involves adopting a mobile and cloud first world, which puts the end user even more in focus. Consequently, fast and highly iterative end user involvement, including prototype testing, becomes a key success factor.

4.3 Prototypes at Microsoft

In general, there are two distinct understandings of the term "prototype." First there is the classical point of view, which sees a prototype as a running technical pre-version of the final product. This means the alpha or beta versions in traditional software development (Budde et al. 1992, p. 52). The interviewees of this case study mentioned that they have to acknowledge this understanding of a prototype when working with the product teams. The second, more common understanding among the participants of this case study, was the usage of artifacts as exploitative prototypes to represent an experience of a product or feature that has not yet been shipped. This includes very low fidelity prototypes such as a wireframe or a sketch on paper as well as high-fidelity functionality of a concept which has not yet been

[7] http://blogs.msdn.com/b/bingdevcenter/

[8] http://www.microsoft.com/en-us/news/ceo/index.html

shipped or has been designed differently from existing features. However, there were two participants who defined a prototype as something beyond a piece of paper and described it as an artifact with which the user can interact. For example, a clickable prototype or something that can generate more interactive feedback; there should be a higher fidelity in the dimension of interaction.

Interviewees mentioned repeatedly (a), the iterative nature and (b) the representation of an experience that helps to validate how well the design meets the user needs, as key parts of a prototype. However, they also mentioned that the definition and understanding of a prototype changes from group to group and is not consistent within the company. Closely connected to the definition is the goal of a prototype. In general, it was stated that prototypes are supposed to support the development process and ideally help to save actual development time. Other goals mentioned include: analyzing and testing a risky design or interaction; visualizing and demonstrating how solutions or features could work; to validate the desirability with end users, developers, and stakeholders; and proof of the technical feasibility usage as specifications for the software development. The last goal is a rare case and only works for specific products, for example in game development, where it is common to use prototypes as specifications instead of written specifications.

4.3.1 Prototype Usage

Prototypes are used throughout all development phases but, depending on the project, in different variations and intensities. Prototypes are often used to a greater extent in the earlier phases than in the later phases, because at a certain point the product development starts. Beyond this point, prototypes stop being explorative and start evolving into the developed product So in the later phases, prototyping is more a part of the implementation process with a more technical focus as opposed to the prototyping of different solutions in the earlier phases.

If necessary, technical prototypes are used to prove the feasibility of a concept. These kinds of prototypes occur primarily in the design phase when possible solutions are evaluated and narrowed down. Sometimes when prototypes are dependent on complex interactions, technical prototypes are necessary to get a rich enough level of feedback. One interviewee mentioned a case, where a developer was involved very early in the process to iteratively provide an interactive explorative prototype. Another use case for technical prototypes is that of usage as specification for the final product. This would be carried out after the requirements analysis and within the design phase of a traditional software lifecycle. Interviewees pointed out that this is a rare case.

In conclusion, prototype usage depends on the product and the related team. Usually, prototypes are used more frequently in the early process stages and the fidelity increases as the project progresses. However, interviewees mentioned rare cases where prototypes are used as specification or no longer after the specifications were defined. The underlying technology to create a prototype is in general not important unless the goal is to prove technical feasibility. For such technical prototypes the technology used becomes much more relevant and is generally close to the technology stack of the final product. For other prototypes the UX researchers or UX designers choose a tool of their preference, to be as productive as possible.

4.4 Research Methodologies

There are two major kinds of methodologies used at Microsoft. One is lab-based research, heavily involving end users, and the other is field research, which is often product unrelated. Interviewees voiced that the methods used are very different depending on the division; because in general the goals and the maturity of the software are different. With new products, researchers do more field and ethnography research to get insights into people's behavior, perception and thoughts in view of their experience with (similar) products.

Once the development is started, researchers need to be very agile to meet the requirements of the product teams and therefore try to work as iteratively as possible and collect feedback very quickly.

4.4.1 Lab-Based Research

According to the interviewees, most of the research is done in a lab-based environment. This includes traditional usability research and exploratory research, (e.g., with structured interviews regarding behavior and studies regarding the information architecture). Classical usability testing is done primarily with prototypes on actual code and products to make sure they fit usability requirements. The lab environment is a classical setup where the test room is connected through a one-way mirror to an observation room. There are two video cameras installed, one capturing the test participant and one the screen. Since people are encouraged to think aloud audio is recorded as well. On rare occasions, eye-tracking data is collected to see which area a participant focuses on. For usability studies, researchers count and note errors including descriptions that are additional to their notes. Furthermore, interviewees state that notes are the most important data source for the evaluation afterwards and that they are sufficient for about 80 %. Sometimes, notes are taken with Morae which is useful in tagging notes and making observations about the user and elements on the screen. One interviewee, described that some researchers print out the screens which are content of the study, and place sticky notes on the printouts with observations about elements on the screens.

A common approach to compare different variations of prototypes or solutions is A/B testing, which is frequently used at Microsoft. If prototypes are flexible, (usually in an early phase), the RITE method (Medlock and Wixon 2002) is used to validate the usability and iterate on prototypes. RITE stands for 'Rapid Iterative Test and Evaluate' and is a variation of classical usability testing, which was developed and documented by Microsoft in 2002. RITE differs from traditional usability testing "by emphasizing extremely rapid changes and verification of the effectiveness of these changes" (Medlock and Wixon 2002).

4.4.2 Field Research

Field research requires different techniques such as inquiries, structured interviews, or observations. Depending on the research question, interviews can be about lives, product usage, or perception thoughts. Often those exploratory research studies do not focus on a specific product, but instead try to research on the general behavior and thoughts of a target group. Therefore, people are observed in their real context for instance at their work place. Another common form of field research at Microsoft is long-term studies. Over the course of the study, which can be from several weeks to months, periodic interviews are scheduled with the participants focusing on their experience with the product.

4.4.3 Quantitative Research

Quantitative research with, for example, surveys is often done in conjunction with qualitative research, conducted either in the field or in a lab environment. Results or insights of a quantitative study are validated through surveys, A/B testing, or other quantitative methods. In case of long-term studies, there is often telemetry data collected in a quantitative way to see how people are using the prototypes or products over time. Therefore, the prototypes have to have a high fidelity and technical basis close to the final product. For instance, data is collected based on how often people use the device, what pages they are visiting, how many times they are sending an email, or how long they are watching movies on a tablet. In addition, the satisfaction over time and reasons for it are determined with help of a linear scale.

4.4.4 Test Participants

At Microsoft end users are strongly integrated into the design and research phases. People try to do testing and validation as often as possible with real end users as the target groups. For usability testing, the interviewees mentioned a sample size of approximately eight to ten participants who are selected out of a large pool of possible participants, either by an internal or external recruiting service. In addition to this, there are usually internal design reviews and tests that are conducted before testing with end users. These design reviews are done with internal colleagues who have not been involved in the design process. As pointed out, the methodologies at Microsoft are mainly based on end user involvement in conjunction with quantitative feedback.

5 Nokia HERE

Nokia HERE is one of three core businesses of the Nokia Corporation and provides location-based intelligence, products, and experiences across multiple screens and operating systems[9]. The HERE business has about 6000 employees in 56 countries worldwide and is not to be confused with the mobile device division of Nokia, which was bought in 2014 by Microsoft Corp. The company name Nokia and HERE will be used in the course of this case study as synonyms and both refer to the HERE business as part of the Nokia Corporation.

Nokia offers products and services mainly for consumers but also provides services to enterprises with a value set that focuses on customers and users. The company builds a large share of its software for mobile devices and the internet. The development of Nokia is organized in small agile teams who are disposed in parallel. On top there is a horizontal product team established with the responsibility for the overall product strategy and to support prototyping activities. In addition to the horizontal product team, there are horizontal design and a research teams, whereby the research team is seen as part of the overall design team (see Fig. 6). In

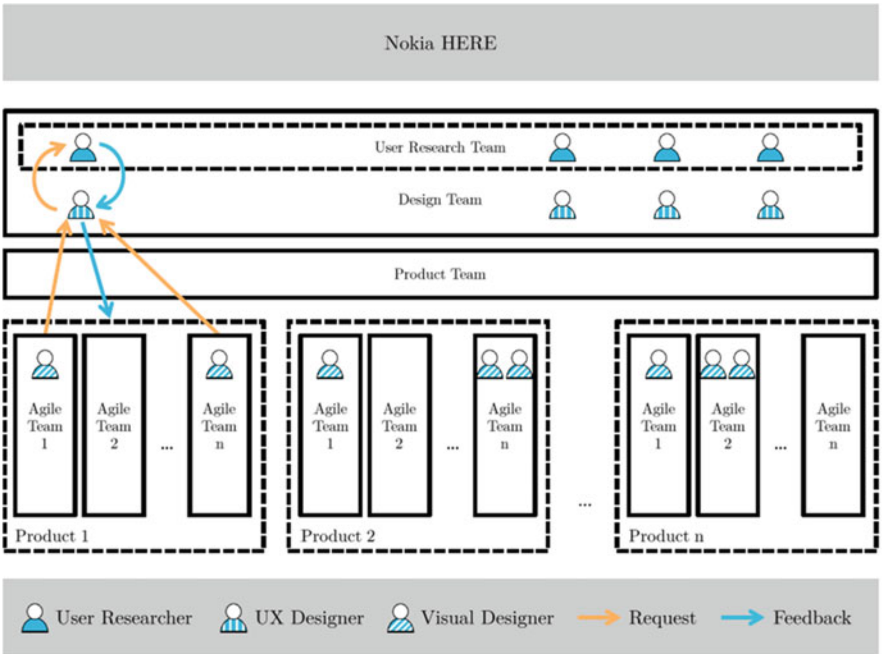


Fig. 6 Embedded roles in the organizational structure of Nokia

[9]http://here.com/abouthere/

fact, the design team is not part of the product teams but dispatches designers to the product teams based on the need. There are numerous product teams that have no contact with the design team and do not require designers since they develop backend, data integration, and other services. The user research team is growing and consists of approximately 15 researchers, who still do not have the capacity to process all requests from the product teams.

5.1 Software Development Roles

There are different technical roles defined for development teams, who follow an agile approach. Typical for some agile methodologies, teams have product owners who are responsible for the success of the product. Of course there are developers who build the actual product. Furthermore, interviewees mentioned the role of a tester, a developer role with the focus on automated testing. For the purpose of this thesis, these different roles are summarized by the term: development team.

In addition to the development teams, there are two roles which concentrate on the user experience of a product. People in the role of user researcher are organized horizontally over the product teams. The user researcher has the main responsibility of developing and executing research studies with a strong focus on end user testing. Researchers organize research activities, coordinate recruiting activities for test participants, and evangelize the importance of end user involvement to the product teams. One important part of their job is to formulate simple tests and to create test scenarios. The second role is called user experience designer. The person in this role is also part of the horizontal design team and works in close collaboration with the research team. User experience designers are often the connecting entity between the product and research team and channel activities regarding user involvement (see Fig. 6). Interviewees describe how they collect the needs of their product team and channel them through to the research team. Moreover, the scenarios for usability testing are written by the researcher and UX designer in collaboration. For smaller testing activities in remote testing (see Sect. 5.4.3) the user experience designer defines and creates test scenarios and researchers assist with support quality assurance at the end.

For the actual design of a product there are visual designers, who are also organized in the horizontal design team. Visual designers are sent to the product team as needed (see Fig. 6). Overall, design and research try to provide support throughout the whole product development cycle, from early conceptual phases to final end-to-end usability testing.

5.2 Development Process

The product development teams at Nokia follow an agile approach that is coordinated with user research and design activities. Individual phases are usually iterative but follow an overall process.

1. First of all, the area and context of the product or feature are investigated. Decisions about the focus for features or about new products come from the strategic horizontal product team. The initial research defining the potential target group is done by the research team or a market intelligence team.
2. Following this step, research about the previously defined target group is done to understand user needs and gather requirements, and to learn about competitors and current ways of solving a problem.
3. Once a rough concept is created and the tasks are prioritized, designers start to create early low-fidelity mockup prototypes. The technology of the prototype is at this stage irrelevant as long as the concept can be visually conveyed.
4. After the basic research, the development team starts to build the product or feature with an agile approach. The design team itself does not operate in an agile mode but adopts the needs to support an agile process through short turnarounds and short testing intervals. However, research and design use the same agile project management tools as development.
5. The previous two steps are highly iterative and involve testing prototypes and pre-versions of the software. Prototypes are adjusted and feedback is pushed back to the development teams.
6. Once the product is close to being shipped, end-to-end usability tests are conducted. When the feature or product is shipped the process starts over again.

5.3 Prototypes at Nokia

The term prototype is used for a wide range of artifacts. Paper-based artifacts of an explorative nature are explicitly seen as prototypes. Additionally, the term prototype also includes pre-versions of a feature or solution built with the final technology stack, which is sometimes referred to as a technical prototype. The technology of a prototype depends on the research question and the goal of the prototype.

5.3.1 Prototype Usage

Prototypes are widely used throughout the entire development process at Nokia and an important factor for product quality as well as user acceptance. According to the interviewees, end user involvement and prototype usage are supported by the management.

Prototypes, except technical prototypes, are mainly used within the design and research team to validate concepts and interactions. In fact, the whole software development process is accompanied by iterative prototyping activities. However, these are in addition to the software development of the actual product. Therefore, prototypes are created iteratively with increasing fidelity but are not designed to be evolutionary prototypes. Nevertheless, a technical feasibility prototype can sometimes evolve into a product and is then tested in iterative, evolutionary ways.

In early phases of a new feature or product, paper prototypes are used to get feedback about ideas and concepts. For these cases researchers print out digital wireframes of visual designs for feedback sessions. Thus the fidelity is higher than for the usual sketches on paper. The importance of using low-fidelity prototypes in early phases to get the right kind of feedback was especially mentioned. By nature, these paper prototypes are not interactive. For testing the interaction in an early stage, tools such as Axure or Balsamiq are used to build interactive wireframes or mockups. In addition, Keynote or PowerPoint were also mentioned as possible tools to build interactive prototypes. In general, interviewees stated the technology is not relevant because they always learn from a prototype as long as the concept is incorporated.

In cases where an early usability study is carried out regarding the interaction with gestures, the Adobe Flash technology is used to develop prototypes which support customized gestures. Long-term and remote studies are conducted with technical prototypes on the final technology stack and sometimes already based on real data.

5.4 Research Methodologies

As part of the product development, research methods mentioned by the interviewees of this case study were mainly based on qualitative research. Before product development, researchers conduct broad general studies about the target user group. Therefore, they use qualitative interviews, focus group discussions, and other open research methods. Most of the research is done in an in-house usability lab.

The research is done very carefully so that every session is analyzed intensively. Thereby, the research team extracts all information from the collected data, organizes it visually on a wall, and decodes the information with colored dots to highlight interesting aspects. During the entire process, researchers try to work openly so that their work is not based on hypothesis. This lends itself to the use of qualitative research methods. Moreover, test participants fill out a questionnaire at the end of each session to measure user satisfaction. To have a large impact on the product, designers and other stakeholders are involved in the process of testing from the beginning instead of only being confronted with the results.

5.4.1 Lab-Based Research

At Nokia two variations of lab-based research are conducted. On the one hand, research studies are based on classical end to end usability tests. On the other hand, Nokia has established a new format called UX carousel. Both types of testing are conducted in Nokias own in-house usability lab. It consists of a testing room and an adjoining observation room connected by a one-way mirror. Besides the test participant, there are usually three more people in the testing room: one researcher leading the interview, one note taker, and one more person who runs the technical equipment. All further people such as the designer, developers, the manager or other stakeholders are responsible for observing the test sessions from the observation room. The testing room is equipped with multiple video cameras and microphones which record everything inside the testing room and stream the video signals in parallel to the observation room. The different video streams from the cameras and the screen are recorded and synchronized with the OVO-Studio[10].

For end to end usability tests, there are test scenarios developed and prepared upfront that are taken from a battery of tests. Each research study has a defined goal and hypothesis about the outcome. In fact, all questions and tasks are included in a test script to cover the same topics with all participants of the same study. The questions and tasks are scenario-based so that the participants are asked to perform the tasks within the given scenario. The analysis of the performance is not based on metrics such as time per task, but rather on observations. This end-to-end testing is primarily done towards the end of a release cycle, when the product or feature is in a beta stadium. The intention is to be able to implement the results and what has been learned from the study. Usually, the number of test participants is between three and five to keep the turnaround short.

The UX carousel is a new format established by Nokia with the intention of reacting quickly to the needs of product teams, who work in an agile process with short release cycles. It takes place every 2 weeks and consists of three 30 min sessions in which a smaller feature is validated each time. Sessions of the UX carousel are scenario-based as well and follow the same procedure as end-to-end usability tests. This format was established because there was a need for designers to validate smaller features in their daily work. Especially features that are added to an existing product often do not justify a complete end-to-end usability study. Interviewees stated that this format works well in an agile development process. Designers need to request a slot 3 to 4 weeks in advance. This is the necessary time to recruit test participants.

Interviewees pointed out that, regardless of the type of study, stakeholders such as designers, developers, business people or managers are always invited to observe the test sessions. This way testing becomes an activity involving the whole team and the acceptance of the results is much higher. All usability sessions are analyzed and evaluated carefully. Therefore, video data is seen as a very powerful and necessary data source to clarify situations that have been perceived differently by different observers. In addition, video, either via streaming or later reply, is necessary to involve and convince other stakeholders of the results.

[10]http://www.ovostudios.com/

5.4.2 Field Research

Nokia conducts general research studies about target groups and user behavior. These kinds of studies are usually not done in a usability lab. For instance, contextual inquiries are conducted in the real context of the participant.

Another form of field test mentioned by the interviewees is a so called snapshot test. During a snapshot test, participants use a prototype in their daily life for a certain period of time. The prototypes for this type of study are already beta versions of a new product. This approach mainly applies to mobile prototype testing.

5.4.3 Remote Testing

In addition to field and lab-based research, Nokia gathers feedback about their products via remote testing. The complete recruiting and testing is thereby outsourced to a specialized provider for remote usability studies. The main advantage is that it is possible to scale the testing easily in terms of quantity of participants. Moreover, Nokia's products target different international markets that can be reached cheaply with this approach. Remote testing was established to better react to the needs of agile development projects that require a fast turnaround for testing. Furthermore, it is meant to become a tool for the daily research into user experience done by designers at Nokia. The intent is to increase end-user involvement in the validation of early prototypes. Remote testing cannot replace moderated testing, but interviewees mentioned two reasons to increase remote usability activities. First, during an agile development phase the need for tests of a feature too small for the UX carousel arises. Second, the user research team is not able to handle all requests for testing. In such cases they still support user experience designers during the task development for remote tests. Due to the fact that the testing is unmoderated, interviewees recommend short test slots from 15 to 30 min. Early visualizations or prototypes are uploaded or provided in another form to be accessed by the participant. In addition, all tasks and questions have to be defined as do criteria for the target users. Within a few hours the test sessions, including video and audio files as well as answered questionnaires, are available for download and can be evaluated by the UX designers.

The main concern regarding remote testing is confidentiality. It was pointed out that researchers need to be very careful with what they share remotely, especially if it is about new ideas or concepts. The concern is not about the legalities of signing a nondisclosure agreement but more about the technical opportunity for participants to record the session. This scenario contrasts to a lab-based test because participants have the prototypes running on their own device and can easily record everything.

5.4.4 Test Participants

At Nokia prototype testing focuses on real end users. Additionally, there are internal colleagues involved in some project phases. Internal validations are usually done in a very early phase of a prototype and are not always carried out in adherence to a strict scientific method. Regardless of the fidelity of the prototype, it is a way for researchers to get an impression of how people would use a new feature or service. This testing is done spontaneously in the canteen and preferably with people who are not part of a product team or involved in design or development activities.

All other usability tests are conducted with end users, who are in the target group of a product. The recruiting of end users is usually done by an external recruiting service which selects users according to a user screening profile. Users for the internal UX carousel (see Sect. 5.4.1) are recruited within 3 weeks, whereas users for end to end usability testing are recruited over longer periods of time. For remote testing, Nokia relies on an external provider who has a comprehensive database of possible test participants. Users are selected on criteria such as age, experience, market, etc. The criteria are chosen by researchers prior to the tests. Shortly before release, if the software prototype is in a beta version, there are sometimes real-life studies conducted. Thereby, the beta version is distributed internally (i.e., employees can download the beta version in an internal app store and test it on their devices in their daily life). Interviewees mention that sometimes such a beta phase study is also conducted with external users.

6 Discussion and Evaluation

Despite the variety of approaches, certain similarities could be found among the case companies. All three companies implement similar roles, which involve prototype creation and testing. All three companies implement a user researcher role, which mainly involves designing and conducting research studies about users. These study may be general or target specific products. The role of a user experience designer, who is responsible for creating prototypes, defining interactions, and designing user interfaces, was also found in all companies. Differences for roles inside the immediate development team were also evident. At Microsoft and Nokia there are mainly two or three roles defined inside a development team, whereas SAP has implemented significantly more specialized roles.

Although the user research and user experience roles bear similar responsibilities inside each company, their integration into the development process is quite different. At Microsoft both roles are deeply integrated into the product teams and are organized in a virtual user experience organization, whereas at SAP and Nokia user researchers have a horizontal organization. Both SAP and Nokia also have specialized visual designers who actually design user interfaces for the final products. At Nokia these designers are part of the product teams, whereas at SAP they are also organized horizontally and design guidelines which are often centrally developed. Thus the individual development teams do not necessarily have immediate access to designers. At all three companies the actual development follows some form of agile software development. However, there are a myriad of different company cultures. On one hand, SAP has a culture of Design Thinking and co-innovation throughout the entire organization. Microsoft's product lifecycle, however, follows a more classical user-centered approach. The UX implementation at Nokia is strongly service-oriented.

6.1 Prototypes

The understanding of the term prototype in the presented case studies is quite different even within the same company. In all three case studies the understanding of a prototype correlates strongly with the employee role. People of the user experience community, such as user researchers, UX designers, and people with a design background, consider artifacts with a very low fidelity to be prototypes. Most interviewees with such a role explicitly declared a sketch on paper to be a prototype. They explained that it helps them to learn more about an area of interest. In contrast, people with a more technical role consider only more advanced artifacts (in particular and sometimes exclusively running software) to be prototypes. This reflects a traditional software engineering point of view.

A general trend towards web-based prototypes could be observed for interactive prototypes. On the one hand, the availability of sophisticated web frameworks makes it easy to build comprehensive prototypes with high fidelity. On the other hand, most online prototyping tools make their prototypes available via HTML.

Interviewees in all case studies stated that the technology of a prototype is only relevant when aiming to prove technical feasibility or for high-fidelity evolutionary software prototypes. In such cases the final technology stack is used.

Interviewees agreed that the choice of tools for prototyping is an individual one. However, for each type of prototype there were certain typical tools mentioned (see Table 1). The main usage of prototypes in all three case companies is for questions of an explorative nature and for traditional usability testing.

6.2 Research Methods

In general, the responsible researchers decide which methods are used for every project individually. Thus, the following discussion is based on general patterns seen within the case companies. Researchers throughout all case companies, pointed out the selection of a research method must be viable in terms of resources and return on investment. Thus, from a professional perspective, the ideal research

Table 1 Comparison of typically mentioned tools for prototype development during the case studies at SAP SE, Microsoft Corp, Nokia HERE

Type of prototype	SAP SE	Microsoft corp.	Nokia HERE
Sketch	Paper	Paper	Paper
Wireframe	Paper, Whiteboard	Paper	Paper
Digital wireframe	Adobe Suite	Adobe Suite	Adobe Suite
Clickable digital wireframe	Axure, Balsamiq, Pidoco, PowerPoint	In Vision, PowerPoint	Axure, Balsamiq, Flash
Clickable fake prototype	HTML, InVision	InVision	Flash
System prototype	Final technology stack, HTML	Final technology stack	Final technology stack
Proof-of-concept implementation	Final technology stack	Final technology stack	Final technology stack

Table 2 Comparison of participants for research studies during the case studies at SAP SE, Microsoft Corp, Nokia HERE

Research goal	SAP SE	Microsoft corp.	Nokia HERE
Ethnographic research	End users	End users	End users
Information architecture	End users, Stakeholders	End users	End users
Early concept validation	Internal colleagues	End users	End users
Concept validation	Stakeholders	End users	End users
Explorative testing	End users	End users	End users
Usability evaluation	End users, Internal experts	End users	End users

is often conducted in a cut down version. Privacy is also an aspect considered for the selection of research methods.

All research methods mentioned can be used for different groups of people such as real end users, colleagues or other stakeholders. Table 2 compares typical test participants for different kinds of research questions between the case companies.

All case companies conduct ethnographic research before the actual design process to learn about general user behavior and target groups. In addition, they use quantitative data collection about existing products or long-term studies. All case companies conduct traditional testing in usability laboratories, usually with some pre-version of the final product. There are two specifics worth mentioning:

- SAP conducts user tests often within a simplified setting in conference rooms or during customer events.
- Microsoft conducts RITE testing (see Sect. 4.4.1), an adapted form of rapid prototyping, inside a lab environment.

Shadowing and contextual inquiries were mentioned by SAP and Microsoft as a form of commonly used field testing methods. Nokia is highly experienced with field testing.

In order to gather feedback and input, SAP does a lot of co-innovation projects with its customers. Moreover, global support and other customer engaging departments push feedback back into the development departments (see Sect. 3.2.2). In contrast, Microsoft and Nokia mainly conduct end-user testing.

Nokia utilizes unmoderated remote testing with a professional provider as a validation tool. SAP uses remote prototype presentations for feedback collection. In contrast, at Microsoft it is neither common to use remote testing nor to send out prototypes to customers.

6.3 Integration of User Research and Prototype Testing into the Development Process

Prototype testing is an integral part of current user-centered approaches. It is a widely used means of improving the quality and user experience of software systems by integrating usability and testing activities into the software development process. Two different approaches can be seen in the case studies.

- SAP and Nokia offer users experience activities through a centralized department organized horizontally over the product teams or the product management.
- At Microsoft, user experience experts are fully integrated into the product team and work in close collaboration with development teams, as is suggested in the literature (Rasmusson 2010, p. 37)

It can be argued that a service-oriented approach has advantages in achieving consistency over different products within a large company and using resources effectively. In fact, this approach is easier and cheaper to implement, especially in large organizations. In practice, however, interviewees of the case companies mentioned critical disadvantages. The UX organizations are not able to handle all requests for user research and UX design support. The process of getting support for UX activities was described as bureaucratic by SAP interviewees. This is critical because it means that development teams—who strive to achieve appealing user interfaces—have a hard time getting the necessary support from experts. SAP has recognized a need for closer collaboration between development and user experience activities and is trying to change its approaches. Interviewees mentioned trials where horizontally organized UX people were integrated into product teams which were seen as very beneficial. Although, user researchers and UX designers are organized horizontally at Nokia, they try to close this gap by involving developers in testing sessions as observers.

In all companies the development process uses an agile framework and aims to follow an agile development approach. The agile manifesto suggests frequent customer involvement (Beck et al. 2001) for such a process. However, it could be observed that usability and end user testing are implemented as additional activities and are not an integral part of the agile development process. Therefore, aligning development cycles and user involvement activities becomes a challenge. Traditional usability studies cannot be easily integrated into the agile process, because research studies are often too comprehensive and require a lot of effort in terms of time and resources. As pointed out by the interviewees of the case studies, in reality agile features are often smaller, and therefore a smaller granularity for testing these is necessary. Moreover, there is a huge mismatch between the cycles of usability testing and agile releases. Thus, it can be pointed out that for agile development the granularity of usability tasks should be small enough to fit into the short release cycles. Sy (Autodesk et al. 2007) also suggests breaking down the design into small design chunks, so-called cycle-sized pieces. Nokia found a good approach to overcome this challenge by conducting user testing in a biweekly cycle in the UX carousel (see Sect. 5.4.1). Furthermore, Nokia is establishing remote usability testing to allow an even smaller granularity of test sessions and implementing testing as everyday tool into the development process.

A deep integration of user testing activities into development teams is seen as beneficial and seen as best practice for software development by UX experts and current research. Consequently, all case companies aim to deepen the integration, which is currently at different levels. Over time this will requires more UX experts within an organization, but as good usability and great user experience become increasingly important, it can be argued that more usability experts are justifiable and would have a return on their investment in the long term (Nielsen 2003).

7 Summary

This chapter presented three case studies on user involvement and prototyping during software development at SAP, Microsoft, and Nokia HERE. As case studies can only provide a small qualitative extract of the overall reality, readers should be aware of their limited general applicability. Moreover, test settings and methodologies are highly dependent on the product and researchers. Consequently, even within one company different approaches and variations of these are used. However certain patterns and trends could be observed.

In general, it was noticed, that the understanding of what a prototype is has changed from that of unfinished, but running, software products to now include low-fidelity prototypes. However, this view still varies greatly among different people. Those in designer roles generally have a broader view, while people in a developer role often have a narrower view. Here, the company culture should try to evangelize the broader view of prototypes to developers.

In our study the research methods at all companies included research before the product development to find needs and understand users in general. Additionally, all companies implemented lab-base as well as field research for usability testing and prototype validation. However, the ratio between these two differed from company to company. SAP and Microsoft used more lab-based and remote testing, while Nokia HERE focused more strongly on field research, which is only a small part of user research and testing at the other two companies.

A general trend to involve more users in design and usability tests with prototypes arises with the switch to agile development. As discussed in Sect. 6.3, this poses a challenge to align the research and testing cycles with the very short and agile development cycles.

References

Autodesk S, Street K, Toronto E (2007) Adapting usability investigations for agile user-centered design. J Usability Stud 2(3):112–132

Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Manifesto for agile software development. http://agilemanifesto.org/. Retrieved 12 Dec 2015

Bright P (2014) How Microsoft dragged its development practices into the 21st century. http://arstechnica.com/information-technology/2014/08/how-microsoft-dragged-its-development-practices-into-the-21st-century/. Retrieved 12 Dec 2015

Budde R, Kauth K, Kuhlenkamp K (1992) Prototyping: an approach to evolutionary system development. Springer, Berlin

Coursaris CK, Kim DJ (2011) A meta-analytical review of empirical mobile usability studies. J Usability Stud 6(3):117–171

Hornbaek K (2006) Current practice in measuring usability: challenges to usability studies and research. Int J Hum-Comp Stud 64(2):79–102

Medlock MC, Wixon D (2002) Using the RITE method to improve products: a definition and a case study. http://www.microsoft.com/usability/Playtest/Publications/Using%20the%20RITE%20Method%20to%20improve%20products.doc;%20a%20definition%20and%20a%20case%20study.do. Retrieved 12 Dec 2015

Microsoft (2013) Annual report 2013. http://www.microsoft.com/investor/reports/ar13/index.html. Retrieved 12 Dec 2015

Nielsen J (1993) Usability engineering: interactive technologies, vol 44. Morgan Kaufmann, Amsterdam

Nielsen J (2003) Return on investment for usability. http://www.nngroup.com/articles/return-on-investment-for-usability/. Retrieved 12 Dec 2015)

Nielsen JJ, Budiu R (2013) Mobile usability. New Riders, Berkeley, CA

Norman DA, Draper SW (1986) User centered system design; new perspectives on human-computer interaction. Lawrence Erlbaum, Hillsdale, NJ

Rasmusson J (2010) The Agile Samurai—how agile masters deliver great software. The Pragmatic Bookshelf

Riley CA, McConkie AB (1989) Designing for usability: human factors in a large software development organization. Conference proceedings, IEEE international conference on systems, man and cybernetics, San Antonio, TX, pp 225–228

Steiner M (2012) Agile methodologies for large enterprises – an analysis of scaling strategies. Hasso-Plattner-Institut, Universität Potsdam

Thomas GS (2012) 16,055 companies fit the definition of large businesses. http://www.bizjournals.com/bizjournals/on-numbers/scott-thomas/2012/07/16055-companies-fit-the-definition-of.html. Retrieved 12 Dec 2015

Part IV
Developing DT Teaching and Coaching Tools and Approaches

Design Thinking At Scale: A Report on Best Practices of Online Courses

Mana Taheri, Thomas Unterholzer, and Christoph Meinel

Abstract Design Thinking has arguably become a state-of-the-art innovation methodology. It has received increasing attention from both media and educational institutes around the globe. Consequently, there is an increasing demand for Design Thinking education. In this research we aim to answer the question of whether and how Design Thinking can be taught in the form of Massive Open Online Courses (MOOCs) that promise scalable teaching. In this chapter we discuss the potentials as well as challenges of teaching Design Thinking in a MOOC environment. In order to learn about the pedagogies and practices required for high quality teaching, we look into four Design Thinking MOOCs and through the lens of a widely used pedagogical framework called the *Seven Principles of Good Practice in Undergraduate Education*. We also pay careful attention to the technological features and the didactical methods applied in selected courses and how they support the fulfillment of these principles. Further, the research team aims at setting up an online course on Design Thinking in collaboration with the openHPI platform—one of the Europe's frontrunner MOOC providers. Thus, we compare the results to the capacities and features of openHPI and examine its potentials for supporting and hosting a design thinking MOOC. Finally based on the best practices observed in the selected courses and the literature, we propose general recommendations for course designers and report on results of interviews with Stanford d.school course instructors on the challenges and potentials of a digital Design Thinking learning environment as well as the path of future research.

M. Taheri (✉) • T. Unterholzer • C. Meinel
Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, Potsdam 14482, Germany
e-mail: mana.taheri@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_13

1 Introduction

Today's complex and intertwined challenges require fresh approaches and skills that guide us towards innovative solutions (Owen 2007). In fact one of the most sought after skills is creative problem solving and the ability to approach problems in novel ways. Design Thinking is undoubtedly among such skills. As a human-centered approach to finding unexpected solutions to complex problems, its popularity is growing steadily. Many educational institutions are either developing a special program around this topic—often along with a dedicated space—or including it in their traditional curriculum.

However, not everyone has access to a world-class Design Thinking education program. Considering the need for innovative solutions to our complex global challenges, and the fact that there are limited opportunities for learning Design Thinking around the world, there is a strong case for teaching Design Thinking at scale and to a broader global audience (Taheri and Meinel 2015).

On the other hand, in the last years the number of institutions opening their lectures to the global public through the internet in a MOOC format has increased. Although MOOCs initially covered conventional knowledge-based courses, as their popularity grew some began to offer courses on trendy and popular topics beyond the traditional university curriculum. Design Thinking was among such topics.

As researchers of the HPI-Stanford Design Thinking Research Program (HPDTRP) at the HPI in Potsdam, where both Design Thinking education and one of Europe's leading MOOC platforms *openHPI* (Meinel and Willems 2013) meet, we saw this development as a great opportunity to investigate the potentials and limitations of teaching and learning Design Thinking at scale—mediated through an open online environment.

In this chapter, we begin by providing a brief review on the development of MOOCs and their role in education today. We then argue that, despite the pre-assumed mismatch between Design Thinking and the MOOC model of learning, in order to create an effective online course on Design Thinking we need to look beyond replicating a real-life experience and instead focus on the learning outcomes and pedagogies. We further report on how Design Thinking is being taught online from an educational perspective. This chapter looks into four MOOCs on the topic that were accessible to us at the time of this study. For this purpose we apply a widely used pedagogical framework called *The Seven Principles of Good Practice in Undergraduate Education* by Chickering and Gamson (1987) to evaluate the selected courses. The framework suggests seven principles that high quality teaching should fulfill. Further, we present some of the best practices seen among the selected courses as well as in other MOOCs and research. In addition, we highlight the features and technological capacities of openHPI in supporting the principles that allow for quality teaching.

Finally, in order to hear the experts' views on the challenges and opportunities of teaching Design Thinking in an online environment, we conducted interviews with experts and course designers from the d.school in Stanford. We discuss the insights gathered from these interviews on the potentials of teaching Design Thinking online in the final section.

1.1 MOOCs: Beyond the Hype

Four years have passed since the *New York Times* proclaimed 2012 “The Year of the MOOC” (Pappano 2012). The article acknowledges that MOOCs had been around for several years as “collaborative techie learning events,” but since 2012 the number of institutes offering online courses around the globe has significantly increased, with that year witnessing a massive upsurge in MOOCs.

On the significant uptake of MOOCs in 2012, Martin Bean, the vice chancellor of Open University states: “In 2012 that wave of disruption hit higher education. By the end of the year, 18 of the top 20 universities in North America were offering MOOCs—so that’s the “great brands” box ticked (Bean 2013).

Parallel to these developments in North America, European companies as well as educational institutions started to build their own MOOC platforms. One of the first European educational/university institutes to offer MOOCs in both German and English on topics related to computer science is HPI, through the openHPI platform (Meinel and Willems 2013).

But what are some of the reasons behind the popularity of MOOCs? We will take a closer look at the major characteristics of MOOCs in the following.

The *massiveness* relates to the potential number of participants in any given course. Unlike the limited number of students attending a course held in a physical lecture hall, with an online course instructors can reach an audience of up to hundreds of thousands from around the globe. Regarding this essential characteristic, Sebastian Thrun, founder of the MOOC platform Udacity, who also offered one of the very first MOOCs, attracting more than 100,000 students, explains that he couldn’t go back to his Stanford classroom after teaching the massive number of students in his Artificial Intelligence MOOC (Hsu 2012).

The word *open* promises a global audience free access to a world-class education. In fact many saw MOOCs as a revolutionary innovation and medium for liberating education by bringing courses taught by world renowned professors to remote villages and economically underprivileged communities around the world.

The most prominent factor regarding the appeal of the term *online* is that as long as there is an internet connection, learning and instruction can happen anywhere and at any time in the world. Thanks to MOOCs, students can now watch course videos in the comfort of their bedroom without struggling to stay focused at eight in the morning in a physical classroom. Because of this significant attribute, some MOOC evangelists even went a step further claiming that the role of universities will become obsolete in the near future.

And last but not least the content is offered in a course format. This means that unlike educational resources, which are freely accessible online (e.g., YouTube videos or MIT's open courseware), in MOOCs the teaching materials are connected and structured within a timeline and with a possibility of receiving a certificate.

A combination of the mentioned attributes and the fact that a number of elite universities joined the trend of offering online courses, led to the hype around the MOOCs and the massive media upsurge. However, despite expectations about MOOCs revolutionizing access to education in deprived communities around the globe, a survey by the University of Pennsylvania in 2013 showed that most of the MOOC users were white males with college degrees (Christensen et al. 2013).

In addition, claims about the disruptive and innovative nature of MOOCs have been challenged. Many (e)learning researchers (e.g., Bates 2015) criticized that treating MOOCs as a completely new phenomenon ignores the prior research in the field of long-distant education. One example is the overemphasized role of video lectures in MOOCs; as Zahn et al. (2014) point out. While in fact peer-to-peer interaction, the key role of students and the connections they form are the components of MOOCs which enable active learning and should be highlighted. Focusing solely on video lectures ignores this potential.

No matter if you are one of the believers or sceptics of the MOOC model of education, one thing is clear: MOOCs are here to stay and are becoming more ubiquitous around the globe. Class Central confirms this notion by showing that the number of students signing up for MOOCs increased from around 18 million in 2014 to 35 million in 2015, reaching an all time high (Shah 2015).

1.2 Design Thinking Meets Video-Centric MOOCs

With the rise in popularity of MOOCs, a number of course designers began to create and offer MOOCs on non-conventional and popular topics including Design Thinking. But can the MOOC model of teaching be effective for courses that are usually not learned individually or are not purely knowledge-based? Consider a typical Design Thinking training for instance. Participants work collaboratively in multidisciplinary teams in an open and flexible space and then go through various hands-on activities and rituals led by a team of facilitators. On the other hand, the picture of an online course participant that comes to one's mind is an individual sitting alone in front of a screen (Taheri and Meinel 2015) (Fig. 1).

The skepticism about the effectiveness of teaching Design Thinking online comes from focusing only on replicating the real life Design Thinking training experience in an online environment. However, such a view neglects the opportunities and potentials that the online world has to offer. The sheer comparison between analog and digital environments doesn't necessarily take into account how well the respective environments were designed. This means whether they have incorporated the latest research and technology or if they use the full and often unique potentials of their medium. With this approach there is no doubt that replicating a collaborative and interactive learning experience of Design Thinking seems impossible in a MOOC format.

Fig. 1 The standard perception of online learning (*left*) versus design thinking training (*right*)

Such comparisons are not only relevant to the field of design. In regard to telecommunication, Hollan and Stornetta (1992) also mention the limitations of research that focuses purely on duplicating the analog experience. They propose instead to "develop tools that people prefer to use even when they have the option of interacting as they have heretofore in physical proximity". According to Hollan and Stornetta "we must develop tools that go beyond being there."

Since the goal of any training is that participants acquire new skills and knowledge, the focus of designing a learning environment—regardless of online or offline—should be on learning outcomes. When investigating the potentials of teaching Design Thinking online, we therefore focus on the pedagogies and learning outcomes rather than replicating the experiences.

2 Research Approach

Putting the skepticism aside, we started to look at the phenomenon of Design Thinking MOOCs from a research perspective. We asked the question whether it is possible to teach Design Thinking online. By the time of this study the popularity of teaching Design Thinking had already stepped into the virtual world and a handful of courses were offered on this topic. Thus, we started our investigation by looking into what is already there and how Design (Thinking) is being taught online. What are some of the practices and approaches applied in the existing courses?

Through a systematic web search we selected four Design (Thinking) courses that were running and accessible to us at the time of this research. We looked into the selected cases from the perspective of participant observers (Bali 2014) and enrolled and engaged in course activities. Since all the team members had some level of experience with both applying and teaching Design Thinking, taking the pure learners perspective was not possible. In other words we could not unlearn what we had learned already. But in order to understand the pedagogies applied in these courses and the technical features behind each course, we engaged in an adequate number of course activities and assignments.

2.1 Theoretical Framework

In order to evaluate the performance of the existing courses regarding their pedagogical practices, we turned to the research field of pedagogy for guidelines. We applied the framework of the *Seven Principles of Good Practice in Undergraduate Education* by Chickering and Gamson (1987), which is one of the most popular and widely used instructional practices stemming from research (Bangert 2004). Chickering and Gamson highlight seven principles that, based on research, effective teaching should fulfill. These principles remain relevant and are used by educators and course designers to assure high quality teaching (Bali 2014). Good practice in undergraduate education has the following attributes:

1. Encouraging contact between the students and faculty
2. Encouraging cooperation among the students
3. Encouraging active learning
4. Providing prompt feedback
5. Emphasizing time on tasks
6. Communicating high expectations
7. Respecting and supporting diverse talents and ways of learning.

Beyond their application in the context of traditional course design, the *Seven Principles* framework translates well into the online education as well as the MOOC teaching model and can guide course designers to create good instructional practices (Siemens and Tittenberger 2009; Bali 2014).

Our aim is to evaluate to what extent the different principles of the framework were fulfilled in the courses and which pedagogical and technological features are behind the best practices. Thus a team of four researchers enrolled in the courses and each reviewer evaluated the performance in the courses regarding these principles. For this purpose we established an arbitrary rating metric: very low, low, medium, high or very high. It is worth noting that these metrics were chosen with the intention of helping us identify the best practices and the technologies and pedagogies behind them. Once all four courses were over, the research team met and discussed the ratings together and as a result an average was created that led to a more objective point of view. In general, the evaluation process was a relatively straightforward one thanks to the simple ranking metrics we chose and in most cases a consensus was reached among the reviewers.

2.2 The Selected MOOCs on Design Thinking

A systematic web search was conducted to identify MOOCs related to the topic of Design Thinking. For this purpose we used multiple MOOC aggregators and the homepages of major MOOC platforms to identify available courses. The result is a selection of four courses that were accessible at the time of the study which are presented in Table 1.

All these courses were offered on an introductory level, requiring no prior knowledge on Design Thinking from the participants. The timeframe of the courses varied from between 4 to 8 weeks and, despite some commonalities, there are some major differences among the courses. For instance, *Design Thinking for Innovative Problem Solving* (DTIP) focuses on the application of Design Thinking in the business domain whereas *Design Kit's* emphasis is on human-centered design for social innovation. The course *Design Creation of Artifacts in Society* focuses on the individual designer's working process in developing human-centered products. Apart from their focus the courses also vary with regard to their pedagogical practices and applied features, which we will discuss in detail in the following chapter.

Besides the abovementioned courses, we identified other courses which are not included in the evaluation. The reason for this is that the two courses of Design Thinking Action Lab (Stanford University) and Innovation and Design Thinking (University of Cincinnati) had concluded with no upcoming iterations being offered. In addition, the course Design Thinking for Business Innovation (Coursera) had just concluded at the time of this study and was therefore not included in our evaluation. However, although not free of charge, another course by the same instructor is included in our study.

Finally in cooperation with openHPI, SAP offers a course called Developing Software Using Design Thinking (DSUDT) on its enterprise MOOC platform, openSAP. Since the course was offered as a pilot and not open to the public it is not included in the investigated sample. However, the research team studied this course closely and had several meetings with the technical and educational course instructors. Since openSAP was developed in cooperation with HPI and they use the

Table 1 A list of the selected MOOCs on Design (Thinking)

Course name	Provider	Platform	Duration	Code
Design thinking online course	Macromedia University	Iversity	4 weeks	DTOC
Design thinking for innovative problem solving	Darden School of Business	NovoEd	8 weeks	DTIP
Design creation of artifacts in the society	University of Pennsylvania	Coursera	8 weeks	DCOA
Design kit: the course for human-centered design	IDEO.org & ACUMEN	NovoEd	7 weeks	DK

same technological features, studying this course helped us to better understand and experience the technological and pedagogical features behind openHPI.

3 Results

Looking at the four selected courses from a pedagogical perspective, using the *Seven Principles* framework, provided us with the clues about how each principle is supported in an online environment. This allowed us to answer the question of whether Design Thinking can be taught with a high pedagogical quality in MOOCs. Table 2 demonstrates the evaluation results. Since the point of this evaluation was to identify the best practices, we decided to omit the course names and refer to the courses using successive numbers.

The results show that all the seven principles have been supported in one way or another in these MOOCs, although the level in which they are fulfilled varies from course to course. Thus the pedagogical principles assuring high quality teaching can be achieved in a MOOC environment. In addition all the principles except for two, namely *providing prompt feedback* and *respecting and supporting diverse talents and ways of learning,* could be fulfilled on a *very high* level based on our evaluation metrics. *Respecting and supporting diverse talents and ways of learning* never received any rating of *high*. This raises the question of whether this principle can be supported on a higher level in a MOOC environment and if yes which pedagogies and features can enable this? Considering the fact that MOOC participants are very diverse regarding their educational and cultural background compared to a typical campus classroom, this principle becomes even more relevant and important, and offers a potential for future research.

Table 2 An overview of the evaluation results based on the framework of *The Seven Principles of Good Practices in Undergraduate Education*

	Course 1	Course 2	Course 3	Course 4
1. Encouraging contact between the students and faculty	Very low	Very high	High	Low
2. Encouraging cooperation among the students	Very low	High	Very high	High
3. Encouraging active learning	Very low	High	Very high	Very high
4. Providing prompt feedback	Low	Medium	Low	High
5. Emphasizing time on tasks	Very low	High	Very high	Low
6. Communicating high expectations	Very low	Very high	Very high	Very high
7. Respecting and supporting diverse talents and ways of learning	Very low	Low	Low	Medium

4 Best Practices

In this chapter we will highlight the best practices seen in the evaluated course and show examples of how all principles can be supported in MOOCs. The courses are presented with their given codes. Furthermore, we discuss the advantages of online courses in comparison with a typical classroom regarding some of these principles. Finally, beyond the four selected courses we report on learnings from other MOOCs and research as well as the potentials of openHPI regarding each principle.

1. Encouraging Student-Faculty Contact At first glance this principle seems very challenging to support in MOOCs. Considering the large number of students attending a given MOOC compared to the limited number of faculty members, effective contact between faculty and the students seems challenging—if not impossible. However some of the courses we observed tried to overcome this hurdle in different ways.

In the DK course, active support for student questions and discussions were provided thanks to an extension of the faculty through the introduction of two roles. The roles were: *course catalysts,* who were alumni of the course and volunteered to use their learnings to support new students, and *teaching assistants*. With the help of these extended faculty members all the students' questions were answered. In addition utilizing the pool of alumni and including them in activities related to supporting students during the course while at the same time offering former students a good chance to put their learnings into practice. On the other hand, the core faculty can take a supervision role and moderate these interactions if needed.

Another best practice was found in the DTIP course itself, which offered live Q&A session in the middle of the course. Here, the students' questions were collected in advance through a forum thread. Adjunct lecturers and community mentors also extended the faculty's presence in the forums. The DTIP course encourages interaction with the faculty on a very high level and has a full time employee to answer questions. It is worth noting that this is a paid course, which results in a lower number of participants (around 200) due to the enrolment fees and in turn makes it easier to manage effective interaction.

Another useful method and feature seen in most of the courses were the weekly emails or video announcements in which the faculty usually encourages participants to pose questions in forums and participate actively in the discussions. In one of the courses some responses to forum questions were contributed and signed by the faculty, which gave credibility to the answers.

Beyond the practices seen in the four selected courses, some MOOCs introduce course-related hashtags (#) for communication and to monitor trending topics through their social media channels. In the course Delft Design Approach (EdX) the most raised questions and topics were collected with the help of related hashtags and instructors could address them in short videos prepared weekly. This gives the impression that the faculty is constantly engaged and is monitoring the course progress instead of having one set of pre-recorded videos for every course iteration. Another common tool that offers the possibility of moving away from standardized videos is Google Hangout. Incorporating at least one Hangout session during the course allows students to get to know the course instructors better and beyond pre-recorded videos.

These examples show that despite the fact that the number of participants poses challenges in achieving the first principle, course designers have displayed creative work-arounds such as recruiting former students, or using common technologies to overcome the hurdles; The former strategy was applied in DSUDT by openSAP, where they created a pool of "coaches" from former course students as well as interested and experienced personnel within SAP.

A relatively recent experience of openHPI offering a course called *Spielend Programmieren* Lernen (playfully learning to program), shows the importance of interaction between faculty and students (Loewis et al. 2015), especially in hands-on and interactive courses that are targeted towards novices or a young audience.

2. Encouraging Cooperation Among Students One of the best practices seen among the selected courses in encouraging student cooperation is the DK course. In this course it is highly recommended that participants form physical teams (i.e., with friends or colleagues). Prior to the start of the lectures there is a week dedicated to forming teams and signing up to join a team.

For those individuals who do not form a team, there is the option of joining already existing teams. However building entirely virtual teams might be problematic since it's required that teams meet up and go through workshop guidelines together to prepare the assignments. Moreover the course administrators provide a map that shows where the participants are located as well as a number of opportunities for in-person meetups in different cities, encouraging participants to create real life communities and stay connected.

Another good example was seen in some courses which require assignments. Here, students are encouraged to review their peers' work and give feedback. The course DCOA enforces peer review by making it mandatory for participants to give feedback on each other's work and penalizes those who do not carry out fair and thorough peer reviews by withdrawing 20 % of their final credits.

In order to spark interaction and exchange among students in the DTIP course the instructor highlights some of the students' posts from the forums in her weekly announcement. Further other students are encouraged to add comments and engage in discussions. On the impact of forum participation, a statistical evaluation of the first two courses at openHPI found that high forum participation directly correlated with better overall results (Gruenewald et al. 2013).

Some examples from other non-Design-Thinking MOOCs include instructors posting questions in forums to initiate discussion threads, encouraging students to seek inspiration by reviewing their peers' works, asking students to upload pictures from their teamwork, and encouraging students to cooperate with peers, at least in certain tasks such as brainstorming.

To facilitate peer interaction one of the solutions from the field of Computer Supported Collaborative Learning (CSCL) that is incorporated in some courses is TalkAbout (Kulkarni et al. 2015b). It connects students to their peers around the world through synchronous video discussions. By forming international discussion teams on the course topics, they also tap into the diversity in the pool of MOOC participants.

A further solution for encouraging peer interaction is forming ad-hoc groups of participants that are online simultaneously working on a group exercise, which takes away the pain of finding a timeslot to meet (Sharples et al. 2014). In general, there is a great opportunity for future research on teamwork and peer interaction in the research fields of CSCL and Computer Supported Collaborative Work (CSCW).

The course DSUDT also applied a practice similar to DK, encouraging participants to form teams. In their latest iteration of the course, they introduced a concept called *week 0*, a week prior to the official start in which participants get acquaintance with the course environment and build teams. According to the course instructors, this leads to a significant increase in course completion.

Moreover, apart from such approaches that can be taken by the instruction team and course designers, from a techonological perspective openHPI offers features that support and facilitate cooperation between students. The platform offers a dedicated space for teams called *collab space*. This space is comprised of forums which can be explicit to team discussions (optional), uploads for different file types, google hangouts, and a virtual whiteboard called *tele-board*. In addition, it contains a group awareness feature that shows the latest status and edits of team members on a participant's dashboard. However, the weak usage of the *collab space* in the first pilots shows a need for improving usability aspects and better integration in the courses (Staubitz et al. 2015).

3. Encouraging Active Learning As Chickering and Gamson put it "Learning is not a spectator sport," in other words high quality teaching requires students to engage in projects, discussions and structured exercises (ebd., 1987). This principle is especially important in teaching Design (Thinking), which applies a project-based approach where students learn "by doing."

Some of the evaluated MOOCs went beyond teaching the process steps and methods and required students to work on a design challenge and apply their learnings. In DCOA, for instance, students had to set up their own homepage using a simple drag-and-drop website creator to upload their weekly assignments and create a portfolio and receive comments and feedback on their work. In DK and DTIP, students worked on a project. The project involved everything from identifying design gaps in everyday life to creating prototypes and uploading the work in progress for feedback. In DK there were weekly workshop guides in a PDF format that guided teams through different activities and tasks. In enforcing project-based learning, submitting all project assignments was a requirement for passing the course, along with completing the course material.

In general, MOOCs with a strong focus on learner-centeredness encourage participants to create their own content and thus step beyond passive video lectures that are common in many current MOOCs. Lloyd (2013) argues that in an online environment, this many-to-many knowledge transmission should be in the center of the course design and be supported by learning activities. This would allow

Fig. 2 Passive video watching (*left*) versus active learning in online courses (*right*)

participants from different backgrounds and expertise to be involved in the problem solving process. Videos on the other hand only add value with the right design (e.g., when embedded into relevant tasks) (Koumi 2015; Pea and Lindgren 2008) (Fig. 2).

Regarding this principle, openHPI is also experienced in offering courses that go beyond sheer video watching and self-assessment tests. Both *Spielend Programmieren Lernen* (see Loewis et al. 2015) and DSDUT have integrated practical tasks and projects in their courses.

4. Providing Prompt Feedback Students need to receive constructive feedback along their learning journey, which also presents them with opportunities for showcasing their progress. In a classroom environment, depending on the number of participants, there are various ways to assess and give feedback. These ways include tests, project results or simple verbal or nonverbal communication. The massive number of participants in MOOCs makes one-on-one instructor feedback an impossible task. That's why MOOC designers need to incorporate new strategies and build on the potentials of digital features that enable prompt feedback and reviews during the course. The most prominent strategies we observed are automated assessments, peer-feedback systems and prompt replies on forum posts.

Peer review was carried out, for example, as part of the students' assessment and final grading. To engage students in reviewing the work of their peers the DCOA course made it obligatory for students to review a certain number of peer works in order to receive feedback on their own work. The instructors provided guidelines and examples of good feedback upfront to assure a fair assessment among peers. In this way all the assignments received some feedback and comments from peers.

Automatic feedback on multiple-choice quizzes and self-assessment questions embedded in videos and course forums are some of the common practices and features applied in MOOCs. Sometimes the faculty members get engaged and respond to some questions in the forums, although due to the number of discussions and questions this is not a scalable solution.

In search of good practices beyond common solutions, it is helpful to look at e-learning for inspiration. For instance, true or false feedback in multiple choice quizzes won't help students to learn about their mistakes and how to improve their abilities, whereas providing hints that help in recognizing the right answer is helpful. This can be achieved by offering a hyperlink to the relevant point in the video lecture or providing additional readings.

In order to facilitate prompt feedback, Kulkarni et al. (2015a) introduced PeerStudio, a platform for peer assessment that taps into the potential of a large number of participants to give feedback on work in-progress. Students who want feedback have to first submit their in-progress work and then give feedback on the work of two peers according to rubrics provided by the instructors. They can then incorporate that feedback into their work and submit again. One of the benefits of this solution is that receiving feedback on work-in-progress allows students to refine and resubmit their work. This is especially useful for design projects as receiving feedback on rapid prototypes can be very helpful for further iterations. Furthermore Kulkarni et al. (2015a) show that in a typical MOOC a median participant received feedback in intervals of no longer than 20 min. They further show the positive impact of prompt feedback on in-progress work using PeerStudio as reflected in students' final grades.

PeerStudio is a great example of a digital solution that is beneficial to both online and physical courses. Because providing feedback on all students' work in a physical classroom is time consuming as well, PeerStudio is able to reduce this time and therefore offers a solution that goes "beyond being there" (see Hollan and Stornetta 1992). At the same time, it is very important that course instructors provide clear guidelines on how to give constructive and fair feedback to peers (Kulkarni et al. 2015a).

In general it is more common to receive feedback on the end results (e.g., product artifacts) than on the learning process, even in a real life settings. Oftentimes the test grades are the main feedback that students receive. These are then relatively general and do not indicate weaknesses and where there's room for improvement Verbal and written comments might solve this problem to some levels. This is also important when providing peer feedbacks.

Regarding this principle, openHPI has also implemented a peer feedback workflow inspired by the existing peer assessment tools. However Its distribution system allows for students who provide more feedback to also get more feedback as a reward (Staubitz et al. 2016).

Experts from the openHPI team suggest that an experienced and highly motivated teaching team can provide for prompt feedback, e.g., through highlighting and linking forum posts with the same topics (using the 'sticky post' feature as mentioned by Staubitz et al. 2015). Further intelligent text-input fields can help to reduce duplicate questions and therefore make the task of providing prompt feedback more achievable.

5. Emphasizing Time on Tasks As Chickering and Gamson (1987) point out, it is important for students to learn to manage their time and allocate an adequate amount of time for effective learning. At the same time, this is also a crucial point for faculty members for effective teaching. Thus it is important to plan the tasks and learning activities with care and to clearly communicate the time needed with students.

Among the reviewed courses DTIP presented a clear guide on the amount of time needed for each task during the 8 week course plan. DK went even one step further and provided micro timing for each step and small tasks in weekly workshop guides. Weekly deadlines for assignment submissions and announcements are some of the common practices seen in many MOOCs.

Whereas the emphasis is usually reached through communication, some courses on openHPI make use of a time-script for assignments that can only be accessed once and need to be finished in a certain time span.

6. Communicating High Expectations Learning a new skill and knowledge requires time and effort regardless if the class is taught online or offline. Thus communicating clear, measurable, and demanding learning outcomes is important. Most of the evaluated courses emphasize that students need to apply their learnings to a real life design challenge in order to learn the methodology. The course DK, for instance, required students submit all four assignments in their design challenge in order to receive the certificate of accomplishment. Incorporating various tasks related to the teaching material sends the signal to students that they need to take an active role in learning—in contrast to the passive consumption of videos.

Many MOOCs seem to shy away from communicating the amount of work and preparation needed to learn the new knowledge and skills and instead keep the learning goals on the level of generic and vague statements. One possible reason is the fear that students might lose motivation and dropout. Examining how to implement meaningful tasks that maintain students' motivation while at same time fulfilling high expectations would be an interesting research topic. Despite these dangers connected to communicating high expectations three courses were rated *Very High*.

Although as the name of this principle suggests it is all about the communication between course providers and the learners, it might be interesting to think of different ways that technological features could support this principle.

7. Respecting and Supporting Diverse Talents and Ways of Working Teaching is not a "one size fits all" activity since individuals learn in many different ways. Respecting and supporting diverse talents and ways of learning is one of the most challenging principles for course instructors to fulfill in real life. It requires them to provide various opportunities and media for their students to choose from. At the same time it is necessary to offer students diverse opportunities to showcase their learning progress.

Many MOOCs, however, tend to put the video lectures in the center of their course, often combining it with multiple choice quizzes and some additional readings. Apart from standard approaches, the course DK provided most of the content in readings combined with visuals in a PDF format. There were short weekly videos, but the main focus was on the readings and workshop guidelines. Moreover, DK offered students various design challenges to choose from: students could either pick one of the three pre-defined challenges offered by the course designers, or identify a challenge from their own social context and apply their learnings to it.

One of the major factors differentiating a MOOC from a campus course is having a massive number of participants from all over the world with diverse cultural and educational backgrounds. Thus apart from supporting the different learning preferences of individuals, language barriers (e.g., non-natives), technology limitations (e.g., internet access) and cultural differences should also be taken into account. In fact while designing a MOOC, designers should consider their global audience and the context in which the course material will be used. For instance creating high quality videos might not be useful for those without a good internet connection and low bandwidth. Another factor in designing a course with a global audience is the language barrier. Avoiding references and examples that are specific to the course providers' culture, including subtitles in the videos or providing a text version of the video lectures are among some of the helpful approaches.

Despite the seemingly challenging nature of this principle, the digital setting seems better suited to treat a massive number of learners individually than a typical classroom. This offers an interesting research opportunity in the field of adaptive learning environments.

Regarding this principle, HPI-run platforms offer solutions such as introducing bonus tasks for those who wish to learn more than the requirements, and different paths and learning journeys for participants to chose from in the course. However incorporating these solutions into a course increases the workload (and eventually the cost) for course designers, as mentioned by the experts.

General MOOC design recommendations derived from the best practices	
Encourage student-faculty interaction • Offer Q&A sessions in the middle of the course to answer the most raised student questions • Extend faculty's appearance by engaging adjunct lecturers and volunteers (e.g., alumni) as moderators *Encourage cooperation among students* • Recommend students to form teams and try to have physical meetings to prepare the assignments • Encourage students to organize in-person meetups • Establish central role of peer review in all assessments • Pick up interesting but unanswered student posts in weekly announcements and encourage others to reply *Encourage active learning* • Let students create their own learning portfolio (e.g., by setting up a personal homepage) • Create a procedural journey with consecutive tasks throughout the weeks	*Provide prompt feedback* • Make peer feedback obligatory for course completion • Apply CSCL/W tools that facilitate peer-review *Emphasize time on task* • Offer (weekly) guidelines that clearly communicate how much time should be allocated per each task • Suggest micro-timing for smaller tasks (e.g., for different brainstorming techniques) *Communicate high expectations* • Communicate your expectations through course videos • Set demanding tasks *Respect diverse talents and ways of learning* • Provide all course materials in a single PDF for those who prefer reading or have limited internet access • Give students options to submit their assignments through different media (e.g., PDF or video) • Provide subtitles • Be mindful of the global audience and

(continued)

General MOOC design recommendations derived from the best practices	
• Provide templates and encourage learners to complete them in teams	potential cultural differences • Integrate bonus tasks or alternative paths

5 Insights from Experts Interviews

In March 2015 we had the chance to conduct interviews with a number of experts involved in MOOC and Design Thinking curriculum-design, both in executive and in student programs at the d.school Stanford. One of the main focuses of the interviews were the challenges and opportunities for teaching Design Thinking online. In this chapter we present a selection of insights that were synthesized from the interviews by a team of four researchers.

Regarding the challenges and limitations of the online world for Design Thinking education, some experts mention the challenge of creating a state of flow in online world, which is very distinctive for real-world Design Thinking trainings. Creating “hooks” and engaging tasks that maintain the audience’s attention in an online scenario might be a solution to this problem. Furthermore, the co-location of team members was mentioned by some experts. One of the interviewees pointed out that being accompanied by another person is crucial in developing empathy, overcoming personal bias and integrating a second perspective. Finally, despite the benefits of diversity among the virtual teams, the difficulties arising from cultural differences in cross-cultural teams were also mentioned in one of the interviews.

Despite the abovementioned challenges, most of the experts saw great opportunities and advantages in an online Design Thinking training. While in a typical training there is one format offered to all participants; online, adaptive learning and individualized training can be achieved. Moreover as learners advance in their learning process and become more proactive, there is a need for a contextualization of their learning process. According to some experts, this need is (left) unfulfilled in structured workshop trainings.

In a typical Design Thinking training, participants go through various activities in a fast pace. Although some use design documentations to capture their main activities, many details get lost in the process. The fact that everything is recorded in a digital environment not only enables the possibility to document what’s going on. It also provides participants a great chance for reflection on their learning journey. An interviewee points out the benefits of peer-learning in an online environment, he described as “crowdsourcing effects.” The instructor compared his experience of teaching a university course with a MOOC he had taught. In the MOOC he had observed a greater number of high-performers willing to take over responsibilities, such as acting as team leaders or moderators, compared to his classroom course. Another opportunity, seen from the instructor’s perspective, is the need to be more precise and concise when giving instructions in an online course. This is due to the lack of live teacher-student interaction. In other words, instructors must make sure that content is understandable and to the point. Some experts stated that their experience preparing a MOOC has enhanced their instruction in the classroom.

Scalability was also mentioned as an asset of the digital Design Thinking education. The practice of scalability empowers individuals with the tools and methods to tackle complex problems in their communities. Finally, as a recommendation it was mentioned that a technology-mediated learning environment should not try to copy the real world but build on its unique possibilities.

6 Summary and Outlook

In this work we have tried to answer the question of whether it is possible to teach Design Thinking effectively in an online environment. We started by looking at MOOCs that are offered on Design Thinking through the lens of the pedagogical framework of the *Seven Principles*, while taking into account the technological features and functionalities behind the course pedagogies. We concluded that although challenging at times, the principles that assure high quality teaching and are used by course designers in real life can also be well satisfied in an online course. Interestingly, some principles can be supported even better in an online environment than in a typical classroom thanks to technological advancement. We further highlighted some of the best practices exhibited in the evaluated courses as well as outstanding examples from research, other MOOCs and experiences from the HPI-run platforms openHPI, openSAP and mooc.house.

In conducting research on learning and teaching Design Thinking online, it is worth noting that although Design Thinking and MOOCs have become popular in recent years, there is a lot to learn from various established research streams such as distant education, pedagogy, and design education.

In this light, in comparing the effectiveness and quality of different courses, it is important to focus on learning outcomes rather than learning experiences. This is also the focus of our future research. In other words, instead of aiming at replicating the experience of a real life Design Thinking training to an online environment, we will investigate the desired learning objectives of Design Thinking education.

For our future work, building on the principles that assure high quality teaching as well as aiming for achieving a set of realistic learning objectives, the research team will create an online course on the topic of design thinking. We have the great chance of collaborating closely with openHPI which, as mentioned above, allows for the fulfilment of the principles required for good teaching, on a relatively high level, with its various technological features as well as a highly experienced team behind the platform. Following a design based research approach we will create interventions and survey acceptance as well as effects on the participants' learning progress. Furthermore, by researching Design Thinking and MOOCs we expect a positive side effects for MOOC research in general in other fields from the perspective of didactics, course design as well as technical support and the feature set. potential new features.

Acknowledgement The authors are greatful for the constructive input from the openHPI experts Jan Renz, Stefanie Schweiger and Thomas Staubitz.

References

Bali M (2014) MOOC pedagogy: gleaning good practice from existing MOOCs. MERLOT J Online Learn Teach 10(1):44–56

Bangert AW (2004) The seven principles of good practice: a framework for evaluating on-line teaching. Internet High Educ 7(3):217–232

Bates AW (2015) MIT and German research on the [appalling] use of video in xMOOCs. URL http://www.tonybates.ca/2015/07/08/mit-and-german-research-on—the-appalling-use-of-video-in-xmoocs/#sthash.Iv5UDe4U.dpuf. Last access on 3 Dec 2015

Bean M (2013) Interview with Martin Bean, Vice-Chancellor the Open University. Future Learn. URL https://about.futurelearn.com/feature/interview-with-martin-bean-vice-chancellor-open-university/. Last access on 10 Jan 2016

Chickering AW, Gamson ZF (1987) Seven principles for good practice in undergraduate education. AAHE Bull 3:7

Christensen G, Steinmetz A, Alcorn B, Bennett A, Woods D, Emanuel EJ (2013) The MOOC phenomenon: who takes massive open online courses and why? Available at SSRN 2350964

Gruenewald F, Mazandarani E, Meinel C, Teusner R, Totschnig M, Willems C (2013) openHPI: a case-study on the emergence of two learning communities. In Global engineering education conference (EDUCON), pp 1323–1331

Hollan J, Stornetta S (1992) Beyond being there. In: Proceedings of the SIGCHI conference on Human factors in computing systems, June, pp 119–125

Hsu J (2012) Professor leaving Stanford for online education startup. NBC News. URL http://www.nbcnews.com/id/46138856/ns/technology_and_science-innovation/t/professor-leaving-stanford-online-education-startup/. Last access on 2 Dec 2015

Koumi J (2015) Learning outcomes afforded by self-assessed, segmented video–print combinations. Cogn Educ 2(1):1045218

Kulkarni C, Bernstein MS, Klemmer S (2015a) PeerStudio: rapid peer feedback emphasizes revision and improves performance. In: Proceedings from the second ACM conference on learning@ Scale, pp 75–84

Kulkarni C, Cambre J, Kotturi Y, Bernstein M, Klemmer S (2015b). Talkabout: making distance matter with small groups in massive classes. In: Proceedings from the 18th ACM conference on computer supported cooperative work & social computing, pp 1116–1128

Lloyd P (2013) Embedded creativity: teaching design thinking via distance education. Int J Technol Des Educ 23(3):749–765

Loewis M, Staubitz T, Teusner R, Renz J, Tannert S, Meinel C (2015) Scaling youth development training in IT using an xMOOC platform. In: Proceedings of the IEEE frontiers in education, El Paso, TX, USA

Meinel C, Willems C (2013) openHPI. The MOOC Offer at Hasso Plattner Institute, Technical Report Nr. 80. University of Potsdam, Hasso-Plattner-Institute for IT-Systems Engineering, Potsdam

Owen C (2007) Design thinking: notes on its nature and use. Des Res Q 2(1):16–27

Pappano L (2012) The year of the MOOC. New York Times. URL http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html. Last access on 1 Dec 2015

Pea R, Lindgren R (2008) Video collaboratories for research and education: an analysis of collaboration design patterns. IEEE Trans Learn Technol 1(4):235–247

Shah D (2015) By the numbers: MOOCS in 2015. How has the MOOC space grown this year? Get the facts, figures, and pie charts. URL https://www.class-central.com/report/moocs-2015-stats/ . Last access on 22 Feb 2016

Sharples M, Adams A, Ferguson R, Gaved M, McAndrew P, Rienties B, Weller M, Whitelock D (2014) Innovating pedagogy 2014. URL http://www.openuniversity.edu/sites/www.openuniversity.edu/files/The_Open_University_Innovating_Pedagogy_2014_0.pdf. Last access on 24 Feb 2016

Siemens G, Tittenberger P (2009) Handbook of emerging technologies for learning. University of Manitoba, Winnipeg, MB

Staubitz T, Pfeiffer T, Renz J, Willems C, Meinel C (2015) Collaborative learning in a MOOC environment. In: Proceedings of the 8th annual international conference of education, research and innovation, pp 8237–8246

Staubitz T, Petrick D, Bauer M, Renz J, Meinel C (2016) Improving the peer assessment experience on MOOC platforms. In: Proceedings of ACM learning at scale conference (L@S2016), Accepted

Taheri M, Meinel C (2015) Pedagogical evaluation of the design thinking MOOCs. In: Proceedings from the 3rd international conference for design education researchers, pp 469–481

Zahn C, Krauskopf K, Kiener J, Hesse F (2014) Designing video for massive open online education: conceptual challenges from a learner-centered perspective. European MOOC Stakeholder's Summit, Lausanne, pp 160–167

Designing Scalable and Sustainable Peer Interactions Online

Chinmay Kulkarni, Yasmine Kotturi, Michael S. Bernstein, and Scott Klemmer

Abstract When students work with peers, they learn more actively, build richer knowledge structures, and connect material to their lives. However, not every peer learning experience online sees successful adoption. This chapter first introduces *PeerStudio*, an assessment platform that leverages the large number of students' peers in online classes to enable rapid feedback on in-progress work. Students submit their draft, give rubric-based feedback on two peers' drafts, and then receive peer feedback. Students can integrate the feedback and repeat this process as often as they desire. PeerStudio demonstrates how rapid feedback on in-progress work improves course outcomes. We then articulate and address three adoption and implementation challenges for peer learning platforms such as PeerStudio. First, peer interactions struggle to bootstrap critical mass. However, class incentives can signal importance and spur initial usage. Second, online classes have limited peer visibility and awareness, so students often feel alone even when surrounded by peers. We find that highlighting interdependence and strengthening norms can mitigate this issue. Third, teachers can readily access "big" aggregate data but not "thick" contextual data that helps build intuitions, so software should guide teachers' scaffolding of peer interactions. We illustrate these challenges through studying 8500 students' usage of PeerStudio and another peer learning platform: Talkabout. Efficacy is measured through sign-up and participation rates and the structure and duration of student interactions. This research demonstrates how large classes can leverage their scale to encourage mastery through rapid feedback and revision, and suggests secret ingredients to make such peer interactions sustainable at scale.

C. Kulkarni (✉) • M.S. Bernstein
Stanford University, Stanford, CA, USA
e-mail: chinmay@cs.stanford.edu

Y. Kotturi • S. Klemmer
University of California, San Diego, CA, USA

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_14

1 Introduction

Many online classes use video lectures and individual student exercises to instruct and assess students. While vast numbers of students log on to these classes individually, many of the educationally valuable social interactions of brick-and-mortar classes are lost: online learners are "alone together" (Turkle 2011).

Social interactions amongst peers improve conceptual understanding and engagement, in turn increasing course performance and completion rates (Porter et al. 2013; Konstan et al. 2014; Kulkarni et al. 2015; Crouch and Mazur 2001; Smith et al. 2009). Benefits aren't limited to the present: when peers construct knowledge together, they acquire critical-thinking skills crucial for life after school (Bransford and Schwartz 1999). Common social learning strategies include discussing course materials, asking each other questions, and reviewing each other's work (Bransford et al. 2000).

However, most peer learning techniques are designed for small classes with an instructor co-present to facilitate, coordinate, and troubleshoot the activity. These peer activities rely on instructors to enforce learning scripts that enable students to learn from the interaction, thus imposing challenges to implementation of peer learning platforms online, at scale (O'Donnell and Dansereau 1995).

How might software enable peer benefits in online environments, where massive scale prevents instructors from personally structuring and guiding peer interactions. Recent work has introduced peer interactions for summative assessment (Kulkarni et al. 2013). How might peer interactions power more pedagogical processes online? In particular, how might software facilitate social coordination?

1.1 Two Peer Learning Platforms

Over the last 2 years, we have developed and deployed two large-scale peer-learning platforms. The first, Talkabout (Fig. 1), brings students in MOOCs together to discuss course materials in small groups of four to six students over Google Hangouts (Cambre et al. 2014). Currently, over 4500 students from 134 countries have used Talkabout in 18 different online classes through the Coursera and Open edX platforms. These classes covered diverse topics: Women's Rights, Social Psychology, Philanthropy, Organizational Analysis, and Behavioral Economics. Students join a discussion timeslot based on their availability, and upon arriving to the discussion, are placed in a discussion group; on average there are four countries represented per discussion group.

We have seen that students in discussions with peers from diverse regions outperformed students in discussions with more homogenous peers, in terms of retention and exam score (Kulkarni et al. 2015). We hypothesize that geographically diverse discussions catalyze more active thinking and reflection. A detailed description in our previous year's chapter in *Design Thinking Research*, discusses

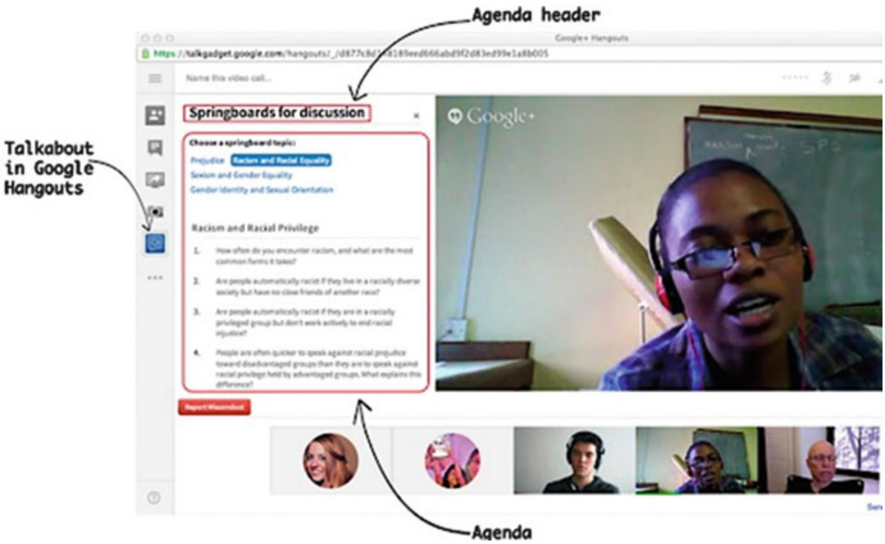


Fig. 1 Our experiences developing and deploying Talkabout, a tool that brings students in massive courses together, helps guide our discussion in the latter part of this chapter

Talkabout's design and pedagogical implications in depth (Plattner et al. 2015). The second platform PeerStudio (Fig. 2) provides fast feedback on in-progress open-ended work, such as essays (Kulkarni et al. 2015). For the implementation and adoption data on PeerStudio, we refer to its use by 4000 students in two courses on Coursera and Open edX. However, we begin with a deep dive into the motives and design for PeerStudio.

1.2 *An In-Depth Look: PeerStudio*

Online learning need not be a loop of watching video lectures and then submitting assignments. To most effectively develop mastery, students must repeatedly revise based on *immediate, focused feedback* (Ericsson et al. 1993). Revision is central to the method of deliberate practice as well as to mastery learning, and depends crucially on rapid formative assessment and applying corrective feedback (Guskey 2007). In domains as diverse as writing, programming, and art, immediate feedback reliably improves learning; delaying feedback reduces its benefit (Kulik and Kulik 1987).

Unfortunately, many learning experiences cannot offer tight feedback-revision loops. When courses assign open-ended work such as essays or projects, it can easily take a week after submission to receive feedback from peers or overworked instructors. Feedback is also often coupled with an unchangeable grade, and classes move to new topics faster than feedback arrives. The result is that many

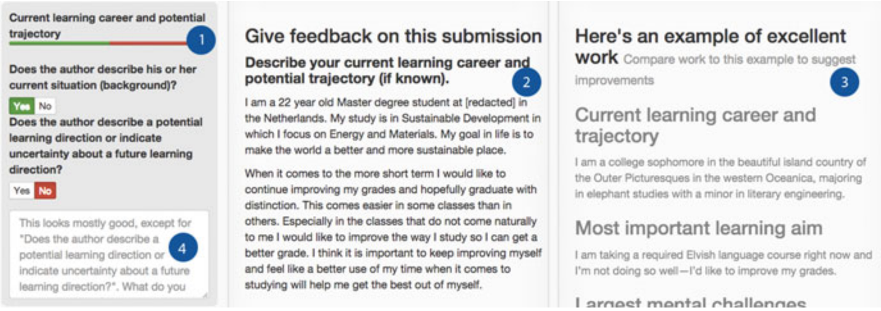


Fig. 2 PeerStudio is a peer learning platform for rapid, rubric-based feedback on drafts. The reviewing interface above shows (1) the rubric, (2) the student draft, (3) an example of excellent work to compare student work against. PeerStudio scaffolds reviewers with automatically generating tips for commenting (4)

opportunities to develop mastery and expertise are lost, as students have few opportunities to revise work and no incentive to do so.

Could software systems enable peers in massive classes to provide rapid feedback on in-progress work? In massive classes, peer assessment already provides summative grades and critiques on final work (Kulkarni et al. 2013), but this process takes days, and is often as slow as in-person classes. This chapter instead introduces a peer learning design tailored for near-immediate peer feedback. It capitalizes on the scale of massive classes to connect students to trade structured feedback on drafts. This process can provide feedback to students within minutes of submission, and can be repeated as often as desired.

We present the *PeerStudio* system for fast feedback on in-progress open-ended work. Students submit an assignment draft whenever they want feedback and then provide rubric-based feedback on two others' drafts in order to unlock their own results. PeerStudio explicitly encourages mastery by allowing students to revise their work multiple times.

Even with the scale of massive classes, there are not always enough students online to guarantee fast feedback. Therefore, PeerStudio recruits students who are online already, and also those who have recently submitted drafts for review but are no longer online. PeerStudio uses a progressive recruitment algorithm to minimize the number of students emailed. It reaches out to more and more students, emailing a small fraction of those who recently submitted drafts each time, and stops recruiting immediately when enough (e.g., two) reviewers have been recruited.

This chapter first reports on PeerStudio's use in two massive online classes and two in-person classes. In a MOOC where 472 students used PeerStudio, reviewers were recruited within minutes (median wait time: 7 min), and the first feedback was completed soon after (median wait time: 20 min). Students in the two, smaller, in-person classes received feedback in about an hour on average. Students took advantage of PeerStudio to submit full drafts ahead of the deadline, and paid particular attention to free-text feedback beyond the explicit rubric.

A controlled experiment measured the benefits of rapid feedback. This between-subjects experiment assigned participants in a MOOC to one of three groups. One control group saw no feedback on in-progress work. A second group received feedback on in-progress work 24 h after submission. A final group received feedback as soon as it was available. Students who received fast in-progress feedback had higher final grades than the control group [$t(98) = 2.1$, $p < 0.05$]. The speed of the feedback was critical: receiving slow feedback was statistically indistinguishable from receiving no feedback at all [$t(98) = 1.07$, $p = 0.28$].

PeerStudio demonstrates how massive online classes can be designed to provide feedback an order of magnitude faster than many in-person classes. It also shows how MOOC-inspired learning techniques can *scale down* to in-person classes. In this case, designing and testing systems iteratively in massive online classes led to techniques that worked well in offline classrooms as well; Wizard of Oz prototyping and experiments in small classes led to designs that work well at scale. Finally, parallel deployments at different scales help us refocus our efforts on creating systems that produce pedagogical benefits at any scale

2 Peerstudio: Related Work

PeerStudio relies on peers to provide feedback. Prior work shows peer-based critique is effective both for in-person (Carlson and Berry 2003; Tinapple et al. 2013) and online classes (Kulkarni et al. 2013), and can provide students accurate numeric grades and comments (Falchikov and Goldfinch 2000; Kulkarni et al. 2013).

PeerStudio bases its design of peer feedback on prior work about how feedback affects learning. By *feedback*, we mean task-related information that helps students improve their performance. Feedback improves performance by changing students' locus of attention, focusing them on productive aspects of their work (Kluger and DeNisi 1996). It can do so by making the difference between current and desired performance more salient (Hattie and Timperley 2007), by explaining the cause of poor performance (Balcazar et al. 1986), or by encouraging students to use a different or higher standard to compare their work against (Latham and Locke 1991).

Fast feedback improves performance by making the difference between the desired and current performance more salient (Kulik and Kulik 1987). When students receive feedback quickly (e.g., in an hour), they apply the concepts they learn more successfully (Kulik and Kulik 1987). In domains like mathematics, computers can generate feedback instantly, and combining such formative feedback with revision improves grades (Heffernan et al. 2012). PeerStudio extends fast feedback to domains such as design and writing where automated feedback is limited and human judgment is necessary.

Feedback merely changes what students attend to, so not all feedback is useful, and some feedback degrades performance (Kluger and DeNisi 1996). For instance, praise is frequently ineffective because it shifts attention *away* from the task and onto the self (Anderson and Rodin 1989).

Therefore, feedback systems and curricular designers must match feedback to instructional goals. Large-scale meta-analyses suggest that the most effective feedback helps students set goals for future attempts, provides information about the quality of their current work, and helps them gauge whether they are moving towards a good answer (Kluger and DeNisi 1996). Therefore, PeerStudio provides a low-cost way of specifying goals when students revise, uses a standardized rubric and free-form comments for correctness feedback, and a way to browse feedback on previous revisions for velocity.

How can peers provide the most accurate feedback? Disaggregation can be an important tool: summing individual scores for components of good writing (e.g. grammar and argumentation) can capture the overall quality of an essay more accurately than asking for a single writing score (Dawes 1979; Kulkarni et al. 2014). Therefore, PeerStudio asks for individual judgments with yes/no or scale questions, and not aggregate scores.

PeerStudio uses the large scale of the online classroom in order to quickly recruit reviewers after students submit in-progress work. In contrast, most prior work has capitalized on scale only after all assignments are submitted. For instance, Deduce It uses the semantic similarity between student solutions to provide automatic hinting and to check solution correctness (Fast et al. 2013), while other systems cluster solutions to help teachers provide feedback quickly (Brooks et al. 2014).

3 Enabling Fast Peer Feedback with Peerstudio

Students can use PeerStudio to create and receive feedback on any number of drafts for every open-ended assignment. Because grades shift students' attention away from the task to the self (Kluger and DeNisi 1996), grades are withheld until the final version.

3.1 Creating a Draft, and Seeking Feedback

PeerStudio encourages students to seek feedback on an initial draft as early as possible. When students create their first draft for an assignment, PeerStudio shows them a minimal, instructor-provided starter template that students can modify or overwrite (Fig. 3). Using a template provides a natural hint for when to seek feedback—when the template is filled out. It also provides structure to students that need it, without constraining those who don't. To keep students focused on the most important aspects of their work, students always see the instructor-provided assignment rubric in the drafting interface (Fig. 3, left). Rubrics in PeerStudio comprise a number of criteria for quality along multiple dimensions.

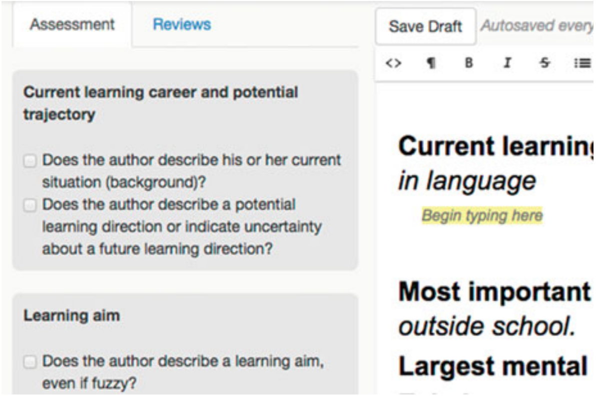


Fig. 3 The drafting interface shows the assignment rubric, and a starter template. Reviews on previous versions are also available (tab, *top-left*)

Students can seek feedback on their current draft at any time. They can focus their reviewers' attention by leaving a note about the kind of feedback they want. When students submit their draft, PeerStudio starts finding peer reviewers. Simultaneously, it invites the student to review others' work.

3.2 *Reviewing Peer Work*

PeerStudio uses the temporal overlap between students to provide fast feedback. When a student submits their draft, PeerStudio asks them to review their peers' submissions in order to unlock their own feedback (André et al. 2012). Since their own work remains strongly activated, reviewing peer work immediately encourages students to reflect (Marsh et al. 1996).

Students need to review two drafts before they see feedback on their work. Reviewing is double blind. Reviewers see their peer's work, student's review request notes, the instructor-created feedback rubric, and an example of excellent work to compare against. Reviewers' primary task is to work their way down the feedback rubric, answering each question. Rubric items are all yes/no or scale responses. Each group of rubric items also contains a free-text comment box, and reviewers are encouraged to write textual comments. To help reviewers write useful comments, PeerStudio prompts them with dynamically generated suggestions.

3.3 *Reading Reviews and Revising*

PeerStudio encourages rapid revision by notifying students via email immediately after a classmate reviews their work. To enable feedback comparison, PeerStudio displays the number of reviewers agreeing on each rubric question, as well as reviewers' comments. Recall that to emphasize iterative improvement, PeerStudio does not display grades, except for final work.

After students read reviews, PeerStudio invites them to revise their draft. Since reflection and goal setting are an important part of deliberate practice, PeerStudio asks students to first explicitly write down what they learned from their reviews and what they plan to do next.

PeerStudio also uses peer assessment for final grading. Students can revise their draft any number of times before they submit a final version to be graded. The final reviewing process for graded submissions is identical to early drafts, and reviewers see the same rubric items. For the final draft, PeerStudio calculates a grade as a weighted sum of rubric items from reviews for that draft.

PeerStudio integrates with MOOC platforms through LTI, which allows students to login using MOOC credentials, and automatically returns grades to class management software. It can be also used as a stand-alone tool.

4 Peerstudio Design

PeerStudio's feedback design relies on rubrics, textual comments, and the ability to recruit reviewers quickly. We outline the design of each.

4.1 Rubrics

Rubrics effectively provide students feedback on the current state of their work for many open-ended assignments, such as writing (Andrade 2001; Andrade 2005), design (Kulkarni et al. 2013), and art (Tinapple et al. 2013). Rubrics comprise multiple dimensions, with cells describing increasing quality along each. For each dimension, reviewers select the cell that most closely describes the submission; in between values and gradations within cells are often possible. Comparing and matching descriptions encourages raters to build a mental model of each dimension that makes rating faster and cognitively more efficient (Gray and Boehm-Davis 2000) (Fig. 4).

When rubric cell descriptions are complex, novice raters can develop mental models that stray significantly from the rubric standard, even if it is shown prominently (Kulkarni et al. 2014). To mitigate the challenges of multi-attribute matching, PeerStudio asks instructors to list multiple distinct criteria of quality along each dimension (Fig. 5). Raters then explicitly choose which criteria are present. Criteria can be binary e.g., "did the student choose a relevant quote that logically supports their opinion?" or scales, e.g., "How many people did the student interview?"

Our initial experiments and prior work suggest that given a set of criteria, raters satisfice by marking some but not all matching criteria (Krosnick 1999). To address

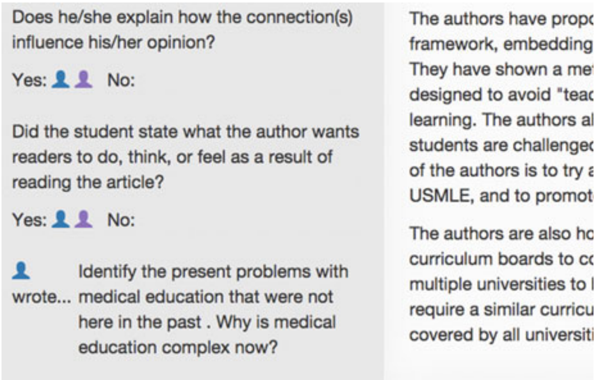


Fig. 4 Students see reviews in the context of their draft (*right*, clipped). PeerStudio displays the number of reviewers (two here) agreeing on each rubric question and comments from each

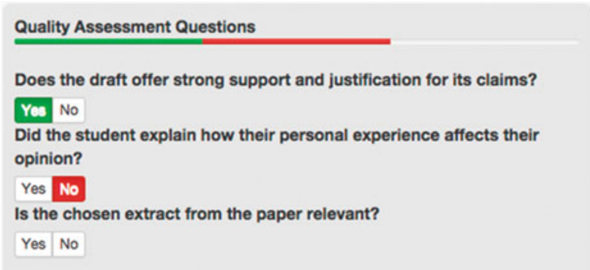


Fig. 5 Example dichotomous questions in PeerStudio. The last question is not yet answered. Students must choose yes/no before they can submit the review

this, PeerStudio displays binary questions as dichotomous choices, so students must choose either yes/no (Fig. 5); and ensures that students answer scale questions by explicitly setting a value.

To calculate final grades, PeerStudio awards credit to yes/no criteria if a majority of reviewers marked it as present. To reduce the effect of outlying ratings, scale questions are given the median score of reviewers. The total assignment grade is the sum of grades across all rubric questions.

4.2 *Scaffolding Comments*

Rubrics help students understand the current quality of their work; free-text comments from peers help them improve it. Reviews with accurate rubric scores, but without comments may provide students too little information.

To scaffold reviewers, PeerStudio shows short tips for writing comments just below the comment box. For instance, if the comment merely praises the submission and has no constructive feedback, it may remind students "Quick check: Is your feedback actionable? Are you expressing yourself succinctly?" Or it may ask reviewers to "Say more. . ." when they write "great job!"

To generate such feedback, PeerStudio compiles a list of relevant words from the student draft and the assignment description. For example, for a critique on a research paper, words like "contribution", "argument", "author" are relevant. PeerStudio then counts the number of relevant words a comment contains. Using this count, and the comment's total length, it suggests improvements. This simple heuristic catches a large number of low-quality comments. Similar systems have been used to judge the quality of product reviews online (Kim et al. 2006).

PeerStudio also helps students provide feedback that's most relevant to the current state of the draft, by internally calculating the reviewer's score for the submission. For a low-quality draft, it asks the reviewer, "What's the first thing you'd suggest to get started?" For middling drafts, reviewers are asked, "This looks mostly good, except for [question with a low score]. What do you suggest they try?" Together, these commenting guides result in reviewers leaving substantive comments.

4.3 Recruiting Reviewers

Because students review immediately after submitting, reviewers are found quickly when there are many students submitting one after another, e.g., in a popular time zone. However, students who submit at an unpopular time still need feedback quickly.

When enough reviewers are not online, PeerStudio progressively emails and enlists help from more and more students who have yet to complete their required two reviews, and enthusiastic students who have reviewed even before submitting a draft. PeerStudio emails a random selection of five such students every half hour, making sure the same student is not picked twice in a 24-h period. PeerStudio stops emailing students when all submissions have at least one review. This enables students to quickly receive feedback from one reviewer and begin revising.

To decide which submissions to show reviewers, PeerStudio uses a priority queue. This queue prioritizes student submissions by the number of reviews (submissions with the fewest, or no, reviews have highest priority), and by the time the submission has been in the review queue. The latest submissions have the highest priority. PeerStudio seeks two reviewers per draft.

5 Field Deployment: In-Person and at Scale

This chapter describes PeerStudio deployments in two open online classes: *Learning How to Learn* (603 students submitting assignments), *Medical Education in the New Millennium* (103 students) on the Coursera and OpenEdX platforms respectively. We also describe deployments in two in-person classes: a senior-level class at the University of Illinois at Urbana-Champaign on *Social Visualization*

(125 students), and a graduate-level class in education at Stanford University, on *Technology for Learners* (51 students).

All four classes used PeerStudio for open-ended writing assignments. In *Learning how to Learn*, for their first assignment students wrote an essay about a learning goal and how they planned to accomplish it using what they learned in class (e.g., one student wrote about being “an older student in Northern Virginia retooling for a career in GIS after being laid off”). In the second assignment, they created a portfolio, blog or website to explain what they learned to others (e.g., one wrote: “I am a professor of English as a Second Language at a community college. I have created a PowerPoint presentation for my colleagues [about spaced repetition and frequent testing]”).

The Social Visualization and Medical Education classes asked students to critique research papers in the area. In Social Visualization, students also used PeerStudio for an open-ended design project on data visualization (e.g., one student team designed a visualization system that used data from Twitter to show crisis needs around the US). Finally, the Technology for Learners class used PeerStudio as a way to critique a learning tool (e.g., ClassDojo, a classroom discipline tool). This class requested its reviewers to sign reviews, so students could follow-up with each other for lingering questions.

5.1 Deployment Observations

Throughout these deployments, we read students’ drafts, feedback, and revisions. We regularly surveyed students about their experiences, and spoke to instructors about their perspectives. Several themes emerged.

5.1.1 Students Requested Feedback on Full Rough Drafts

Rather than submit sections of drafts, students submitted full rough drafts. Drafts were often missing details (e.g., lacking examples). In the Medical Education critique, one question was “did you find yourself mostly agreeing or mostly disagreeing with the content of the research paper? Why?” In initial drafts, students often pointed out only one area of disagreement, later drafts added the rest. Other drafts were poorly explained (e.g., lacking justification for claims) or too rambling.

Students typically asked for four kinds of feedback: (1) On a specific aspect of their work, e.g., “I guess I need help with my writing, vocabulary and grammar, since I’m not an English native-speaker”; (2) On a specific component of the assignment: e.g., “Can you let me know if part 4 and 5 make sense—I feel like I am trying to say too much all in one go!” (3) As a final check before they turned in their work: e.g., “This draft is actually a ‘release candidate’. I would like to know if I addressed the points or if I missed something.” (4) As a way to connect with classmates: e.g., “I just want to hear about your opinions”.

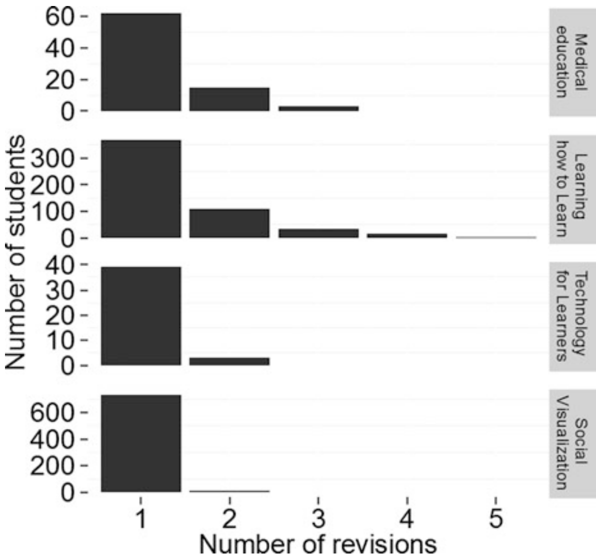


Fig. 6 Most students created a single revision. Students in MOOCs revised more than students in in-person classes

When students revised their draft, we asked, "Overall, did you get useful feedback on your draft?" as a binary question—80 % answered 'yes'.

5.1.2 Students Revise Rarely, Especially in In-Person Classes

Most students did not create multiple drafts (Fig. 6). Students in the two MOOCs were more likely to revise than students in in-person classes [$t(1404) = 12.84$, $p < 0.001$]. Overall, 30.1 % of online students created multiple revisions, but only 7 % of those in in-person classes did.

When we asked TAs in the in-person classes why so few students revised, they told us they did not emphasize this feature of PeerStudio in class. Furthermore, student responses in surveys indicated that many felt their schedule was too busy to revise. One wrote it was unfair to "expect us to read some 40 page essays, then write the critiques and then review two other people, and then make changes on our work… twice a week." These comments underscore that merely creating software systems for iterative feedback is not enough—an iterative approach must be reflected in the pedagogy as well.

5.1.3 Students See Comments as More Useful Than Rubric Feedback

Students could optionally rate reviews after reading them and leave comments to staff. Students rated 758 of 3963 reviews. We looked at a random subset of 50 such comments. In their responses, students wrote that freeform comments were useful (21 responses) more often than rubric-based feedback (5 responses). Students also disagreed more with reviewers' comments (7 responses) than with their reviewers' marked rubric (3 responses). This is possibly because comments can capture useful interpretive feedback, but differences in interpretation lead to disagreement.

An undergraduate TA looked at a random subset of 150 student submissions, and rated reviewer comments on a 7-point Likert scale on how concretely they helped students revise. For example, here is a comment that was rated "very concrete (7)" on an essay about planning for learning goals:

> "What do you mean by 'good schedule'? There's obviously more than one answer to that question, but the goal should be to really focus and narrow it down. Break a larger goal like "getting a good schedule" into concrete steps such as: 1) get 8 h of sleep, 2)...

We found 45 % of comments were "somewhat concrete" (a rating of 5 on the scale) or better, and contained pointers to resources or specific suggestions on how to improve; the rest of the comments were praise or encouragement. Interestingly, using the same 7-point Likert scale, students rated reviews as concrete more often than the TA (55 % of the time).

Students reported relying on comments for revising. For instance, the student who received the above comment wrote, "I somehow knew I wasn't being specific... The reviewer's ideas really helped there!" The lack of comments was lamented upon, "The reviewer did not comment any feedback, so I don't know what to do."

One exception to the general trend of comments being more important was students who submitted 'release candidate' drafts for a final check. Such students relied heavily on rubric feedback: "I have corrected every item that needed attention to. I now have received all yes to each question. Thanks guys. :-)"

5.1.4 Comments Encourage Students to Revise

The odds of students revising their drafts increase by 1.10 if they receive any reviews with free-form comments ($z = 4.6$, $p < 0.001$). Since fewer than half the comments contained specific improvement suggestions, this suggests that, in addition to being informational, reviewer comments also play an important motivational role.

5.1.5 Revisions Locally Add Information, Improve Understandability

We looked at the 100 reflections that students wrote while starting the revision to understand what changes they wanted to make. A majority of students (51 %) intended to add information based on their comments, e.g., "The math teacher [one of the reviewers] helped me look for other sources relating to how math can be fun and creative instead of it being dull!" A smaller number (16 %) wanted to change how they had expressed ideas to make them easier to understand, e.g., "I did not explain clearly the three first parts... I shall be clearer in my re-submission" and, “I do need to avoid repetition. Bullets are always good.” Other changes included formatting, grammar, and occasionally wanting a fresh start. The large fraction of students who wanted to add information to drafts they previously thought were complete suggests that peer feedback helps students see flaws in their work, and provides new perspectives.

Most students reworked their drafts as planned: 44 % of students made substantive changes based on feedback, 10 % made substantive changes not based on the comments received, and the rest only changed spelling and formatting. Most students added information to or otherwise revised one section, while leaving the rest unchanged.

5.2 *PeerStudio Recruits Reviewers Rapidly*

We looked at the PeerStudio logs to understand the platform’s feedback latency. Reviewers were recruited rapidly for both in in-person and online classes (see Fig. 7), but the scale of online classes has a dramatic effect. With just 472 students using the system for the first assignment in Learning How to Learn, the median recruitment-time was 7 min and the 75th quartile was 24 min.

5.2.1 Few Students Have Long Wait Times

PeerStudio uses a priority queue to seek reviews; it prioritizes newer submissions given two submissions with the same number of reviews. This reduces the wait time

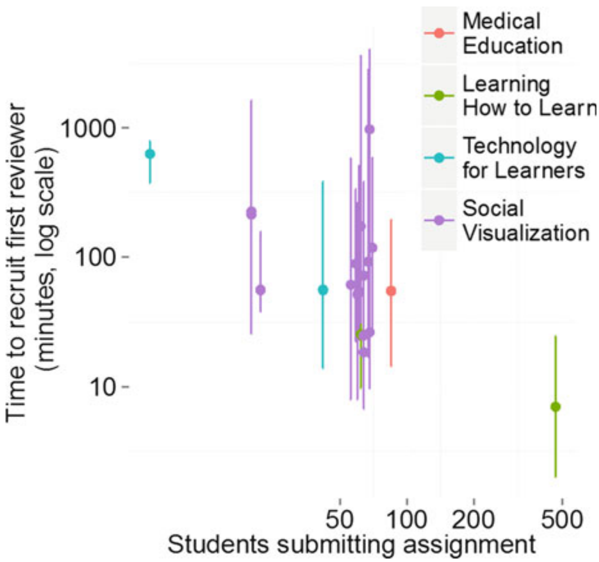


Fig. 7 Reviewers are recruited faster in larger classes

for the *average* student, but unlucky students have to wait longer (e.g. when they submit just before a popular time, and others keep submitting newer drafts). Still, significant delays are rare: 4.4 % had no reviews in the first 8 h; 1.8 % had no reviews in 24 h. To help students revise, staff reviewed submissions with no reviews after 24 h.

5.2.2 Feedback Latency Is Consistent Even Early in the Assignment

Even though fewer students use the website farther from the deadline, peer review means that the workload and review labor automatically scale together. We found no statistical difference in recruitment time [$t(1191) = 0.52$, $p = 0.6$] between the first two and last two days of the assignment, perhaps because PeerStudio uses email to recruit reviewers.

5.2.3 Fewer Reviewers Recruited Over Email with Larger Class Size

PeerStudio emails students to recruit reviewers only when enough students aren't already on the website. In the smallest class with 46 students submitting, 21 % of reviews came from Web solicitation and 79 % of reviews were written in response to an emailed request. In the largest, with 472 students submitting, 72 % of reviews came from Web solicitation and only 28 % from email (Fig. 8). Overall, students responded to email requests approximately 17 % of the time, independent of class size.

These results suggest that PeerStudio achieves quick reviewing in small, in-person classes by actively bringing students online via email, and that this becomes less important with increasing class size, as students have a naturally overlapping presence on site.

5.2.4 Reviewers Spend About 10 min Per Draft

PeerStudio records the time between when reviewers start a review and when they submit it. In all classes except the graduate level Technology for Learners, students spent around 10 min reviewing each draft (Fig. 9). The median reviewer in the graduate Technology for Learners class spent 22 min per draft. Because all students in that class started reviewing in-class but finished later, its variance in reviewing times is also much larger.

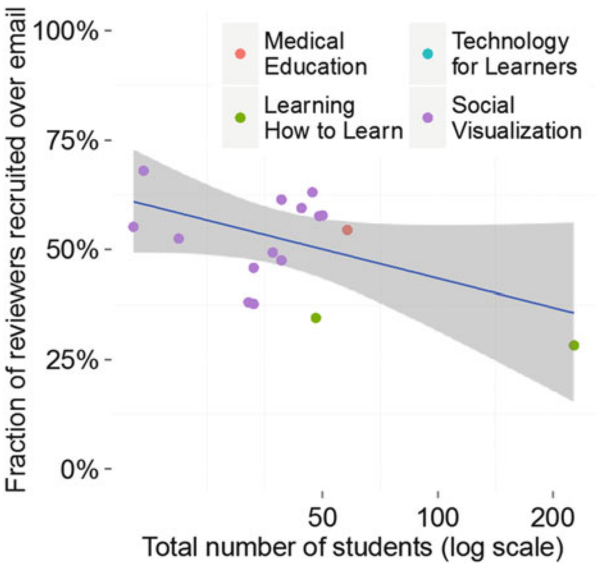


Fig. 8 More students in large classes are likely to be online at the same time, so fewer reviewers were recruited by email

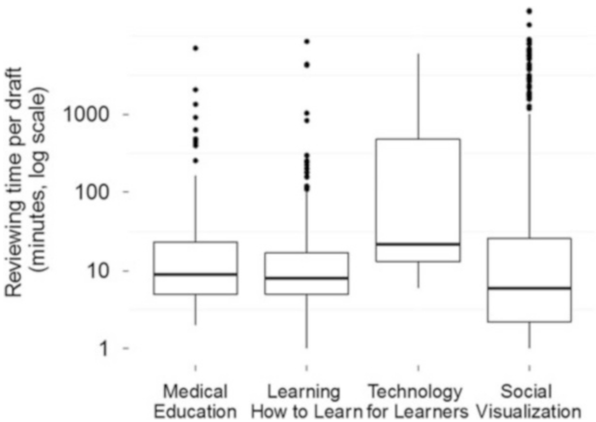


Fig. 9 Reviewers spend roughly 10 min reviewing each draft. The graduate-level Technology for Learners class spends longer. (The larger variation is because students start reviewing in class, and finish later.)

5.3 *Are Reviewers Accurate?*

There is very strong agreement between individual raters while using the rubric. In online classes, the median pair-wise agreement between reviewers on a rubric question is 74 %, while for in-person classes it is 93 %. However, because most drafts completed a majority of the rubric items successfully, baseline agreement is high, so Fleiss' k is low. The median k = 0.19 for in-person classes, and 0.33 for online classes, conventionally considered "Fair agreement". In in-person classes, on average staff and students agreed on rubric questions 96 % of the time.

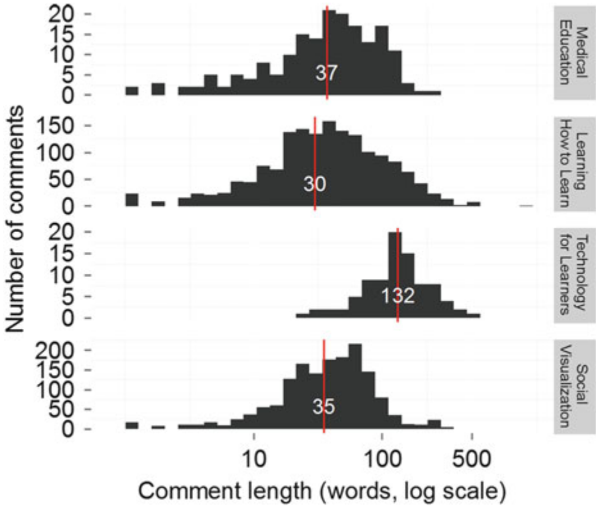


Fig. 10 Students write substantive comments, both in-person and online. The graduate level Technology for Learners has longer comments, possibly because reviews were signed

5.4 *Staff and Peers Write Comments of Similar Length*

Both in-person and online, the median comment was 30 words long (Fig. 10). This length compares well with staff comments in the *Social Visualization* class, which had a median of 35 words. Most reviews (88 %) had at least some textual comments, in addition to rubric-based feedback.

5.5 *Students Trade-Off Reviewing and Revising*

23 % of students reviewed more than the required two drafts. Survey results indicated that many such students used reviewing as an inexpensive way to make progress on their own draft. One student wrote that in comparison to revising their own work, "being able to see what others have written by reviewing their work is a better way to get feedback." Other students reviewed peers simply because they found their work interesting. When told she had reviewed 29 more drafts than required, one student wrote, "I wouldn't have suspected that. I kept reading and reviewing because people's stories are so interesting."

5.6 *Students Appreciate Reading Others' Work More Than Early Feedback and Revision*

A post-class survey in *Technology For Learners* asked students what they liked most about PeerStudio (30 responses). Students most commonly mentioned (in 13 responses) interface elements such as being able to see examples and rubrics. Reading each other's work was also popular (8 responses), but the ability to revise was rarely mentioned (3 responses). This is not surprising, since few students revised work in in-person classes.

Apart from specific usability concerns, students' most frequent complaint was that PeerStudio sent them too much email. One wrote, "My understanding was that students would receive about three, but over the last few days, I've gotten more." Currently, PeerStudio limits how frequently it emails students; future work could also limit the total number of emails a student receives.

6 Field Experiment: Does Fast Feedback on In-Progress Work Improve Final Work?

The prior study demonstrated how students solicited feedback and revised work, and how quickly they can obtain feedback. Next, we describe a field experiment that asks two research questions: First, does feedback on in-progress work improve student performance? Second, does the speed of feedback matter? Do students perform better if they receive rapid feedback?

We conducted this controlled experiment in *ANES 204: Medical Education in the New Millennium*, a MOOC on the OpenEdX platform.

Students in this class had working experience in healthcare professions, such as medical residents, nurses and doctors. In the open-ended assignment, students read and critiqued a recent research paper based on their experience in the healthcare field. For example, one critique prompt was "As you read, did you find yourself mostly agreeing or mostly disagreeing with the content? Write about three points from the article that justify your support or dissent." The class used PeerStudio to provide students both in-progress feedback and final grades.

6.1 Method

A between-subjects manipulation randomly assigned students to one of three conditions. In the *No Early Feedback* condition, students could only submit one final draft of their critique. This condition generally mimics the status quo in many classes, where students have no opportunities to revise drafts with feedback. In the *Slow Feedback* condition, students could submit any number of in-progress drafts, in addition to their final draft. Students received peer feedback on all drafts, but this feedback wasn't available until 24 h after submission. Additionally, students were only emailed about their feedback at that time. This condition mimics a scenario where a class offers students the chance to revise, but is limited in its turnaround time due to limited staff time or office hours. Finally, in the *Fast Feedback* condition, students could submit drafts as in the slow feedback condition, but were shown reviews as soon as available, mirroring the standard PeerStudio setup.

Students in all conditions rated their peers' work anonymously; reviewers saw drafts from all conditions and rated them blind to condition. Our server introduced all delays for the Slow Feedback condition after submission. Rubrics and the interface students used for reviewing and editing were identical across conditions.

6.2 Measures

To measure performance, we used the grade on the final assignment submission as calculated by PeerStudio. Since rubrics only used dichotomous questions, each rubric question was given credit if a majority of raters marked "yes". The grade of each draft was the sum of credit across all rubric questions for that draft.

6.3 Participants

In all, 104 students participated. Of these, three students only submitted a blank essay; their results were discarded from analysis. To analyze results, we built an ordinary-least-squares regression model with the experimental condition as the predictor variable, using *No Early Feedback* as the baseline ($R^2 = 0.02$).

6.4 Manipulation Check

While PeerStudio can provide students feedback quickly, this feedback is only useful if students actually read it. Therefore, we recorded the time students first read their feedback. The median participant in the *Fast Feedback* condition read their reviews 592 min (9.8 h) after submission; the median for the Slow Feedback condition was 1528 min (26.6 h). This suggests that the manipulation effectively delayed feedback, but the difference between conditions was more modest than planned.

6.5 Results: Fast Early Feedback Improves Final Grades

Students in the *Fast Feedback* condition did significantly better than those in *No Early Feedback* condition ($t(98) = 2.1$, $p < 0.05$). On average, students scored higher by 4.4 % of the assignment's total grade: i.e., enough to boost a score from a B+ to an A−.

6.5.1 Slow Early Feedback Yields No Significant Improvement

Surprisingly, we found that students in the *Slow Feedback* condition did not do significantly better than those in the *No Early Feedback* condition [$t(98) = 1.07$, $p = 0.28$]. These results suggest that for early feedback to improve student performance, it must be delivered quickly.

Because of the limited sample size, it is also possible this experiment was unable to detect the (smaller) benefits of delayed early feedback.

6.5.2 Students with Fast Feedback Don't Revise More Often

There was no significant difference between the number of revisions students created in the *Fast* and *Slow feedback* conditions [$t(77) = 0.2$, $p = 0.83$]: students created on average 1.33 drafts; only 22 % of students created multiple revisions. On average, they added 83 words to their revision, and there was no significant difference in the quantity of words changed between conditions [$t(23) = 1.04$, $p = 0.30$].

However, students with *Fast feedback* referred to their reviews marginally more frequently when they entered reflections and planned changes in revision [$\chi^2(1) = 2.92$, $p = 0.08$]. This is consistent with prior findings that speed improves performance by making feedback more salient.

Even with only a small number of students revising, the overall benefits of early feedback seem sizeable. Future work that better encourages students to revise may further increase these benefits.

7 Discussion

The field deployment and subsequent experiment demonstrate the value of helping students revise work with fast feedback. Even with a small fraction of students creating multiple revisions, the benefits of fast feedback are apparent. How could we design pedagogy to amplify these benefits?

7.1 Redesigning Pedagogy to Support Revision and Mastery

In-person classes are already using PeerStudio to change their pedagogy. These classes did not use PeerStudio as a way to reduce grading burden: both classes still had TAs grade every submission. Instead, they used PeerStudio to expose students to each other's work and to provide them feedback faster than staff could manage.

Fully exploiting this opportunity will require changes. Teachers will need to teach students about when and how to seek feedback. Currently, PeerStudio encourages students to fill out the starter template before they seek feedback. For some domains, it may be better to get feedback using an outline or sketch, so reviewers aren't distracted by superficial details (Sommers 1982). In domains like design, it might be useful to get feedback on multiple alternative designs (Dow et al. 2011). PeerStudio might explicitly allow these different kinds of submissions.

PeerStudio reduces the time to get feedback, but students still need time to work on revisions. Assignments must factor this revision time into their schedule. We find it heartening that 7 % of in-person students actually revised their drafts, even when their assignment schedules were not designed to allow it. That 30 % of online students revised assignments may partly be because schedules were designed around the assumption that learners with full-time jobs have limited time: consequently, online schedules often provide more time between assignment deadlines.

Finally, current practice rewards students for the final quality of their work. PeerStudio's revision process may allow other reward schemes. For instance, in domains like design where rapid iteration is prized (Buxton 2007; Dow et al. 2009), classes may reward students for sustained improvement.

7.2 Plagiarism

Plagiarism is a potential risk of sharing in-progress work. While plagiarism is a concern with all peer assessment, it is especially important in PeerStudio because the system shares work before assignments are due. In classes that have used PeerStudio so far, we found one instance of plagiarism: a student reviewed another's essay and then submitted it as their own. While PeerStudio does not detect plagiarism currently, it does record what work a student reviewed, as well as every revision. This record can help instructors check that the work has a supporting paper trail. Future work could automate this.

Another risk is that student reviewers may attempt to fool PeerStudio by giving the same feedback to every assignment they review (to get past the reviewing hurdle quickly so they can see feedback on their work). We observed three such instances. However, 'shortcut reviewing' is often easy to catch with techniques such as inter-rater agreement scores (Kittur et al. 2008).

7.3 Bridging the In-Person and At-Scale Worlds

While it was designed for massive classes, PeerStudio "scales down" and brings affordances such as fast feedback to smaller in-person classes. PeerStudio primarily relies on the natural overlap between student schedules at larger scales, but this overlap still exists at smaller scale and can be augmented via email recruitment.

PeerStudio also demonstrates the benefits of experimenting in different settings in parallel. Large-scale between-subjects experiments often work better online than in-person because in-person, students are more likely to contaminate manipulations by communicating outside the system. In contrast, in-person experiments can often be run earlier in software development using lower-fidelity approaches and/or greater support. Also, it can be easier to gather rich qualitative and observational data in person, or modify pilot protocols on the fly. Finally, consonant results in in-person and online deployments lend more support for the fundamentals of the manipulation (as opposed to an accidental artifact of a deployment).

7.4 Future Work for PeerStudio

Some instructors we spoke to worried about the overhead that peer assessment entails (and chose not to use PeerStudio for this reason). If reviewers spend about 10 min reviewing work as in our deployment, peer assessment arguably incurs a 20-min overhead per revision. On the other hand, student survey responses indicate that they found looking at other students' work to be the most valuable part of the assessment process. Future work could quantify the benefits of assessing peer work, including inspiration, and how it affects student revisions. Future work could also reduce the reviewing burden by using early reviewer agreement to hide some rubric items from later reviewers (Kulkarni et al. 2014).

7.4.1 Matching Reviewers and Drafts

PeerStudio enables students to receive feedback from peers at any time, but their peers may be far earlier or more advanced in their completion of the assignment. Instead, it may be helpful to have drafts reviewed by students who are similarly advanced or just starting. Furthermore, students learn best from examples (peer work) if they are approachable in quality. In future work, the system could ask or learn the rough state of the assignment, and recruit reviewers who are similar.

8 Sustainable Peer Interactions: Three Adoption Challenges and Solutions

With an in depth look into PeerStudio's motives and design, we can better understand the implementation and adoption challenges surrounding peer learning platforms. In this second portion of the chapter, we discuss three such challenges, that have consistently recurred as we have introduced peer learning (PeerStudio and Talkabout) into massive online classes.

First, many courses falsely assume that students will naturally populate the peer learning systems in their classes: "build it and they will come". This assumption often seems natural; after all, students naturally engage with social networks such as Facebook and Twitter. However, students don't yet know why or how they should take advantage of peer learning opportunities. Peer learning platforms sit not in a social setting, but in an educational setting, which has its own logic of incentives: both carrots and sticks are required to keep the commons vibrant. Participation in educational settings has a different incentive structure than a socialization setting. In particular, the benefits of participation are not immediately apparent. For instance, many American college graduates retrospectively credit their dorms as having played a key role in their social development (Dourish and Bell 2007). Yet, universities often have to require that freshmen live in the dorms to ensure the joint experience. We encourage instructors to take a similar reinforcing approach online: integrating peer-learning systems into the core curriculum and making them a required or extra-credit granting part of the course, rather than optional "hang-out" rooms.

The second challenge is that students in online classes lack the ambient social encouragement that brick and mortar settings provide (Erickson and Kellogg 2000). The physical and social configurations of in-person schools (especially residential ones) offer many opportunities for social encouragement (Crouch and Mazur 2001; Dourish and Bell 2007). For example, during finals week, everyone else is studying too. However, other students' activity is typically invisible online, so students do not receive the tacit encouragement of seeing others attend classes and study (Greenberg 2009; Dourish and Bellotti 1992). We hypothesize that in the minimal social context online, software and courses must work especially hard to keep students engaged through highlighting codependence and strengthening positive norms.

The third challenge we have encountered is that instructors can, at best, observe peer interactions through a telescope clouded by big data exhaust: there are few visible signals beyond engagement (e.g. course forum posts and dashboards) and demographics. Student information is limited online (Stephens-Martinez et al. 2014), and knowing how to leverage what demographics instructors do know is non-obvious. In-person, instructors use a lot of information about people to structure interactions (Rosenberg et al. 2007). For example, instructors can observe and adapt to student reactions while facilitating peer interactions. The lack of information in online classes creates both pedagogical and design challenges (Kraut et al. 2012). For instance, in an online discussion, do students completely ignore the course-related discussion prompts and, instead, talk about current events or pop culture? To address such questions, teachers must have the tools to enable them to learn how to scaffold peer interactions from behind their computers.

This half of the chapter addresses these three logistical and pedagogical challenges to global-scale peer learning (Fig. 11). We suggest socio-technical remedies that draw on our experience with two social learning platforms—Talkabout and PeerStudio—and with our experience using peer learning in the classroom. We report on these challenges with both quantitative and qualitative data. Quantitative

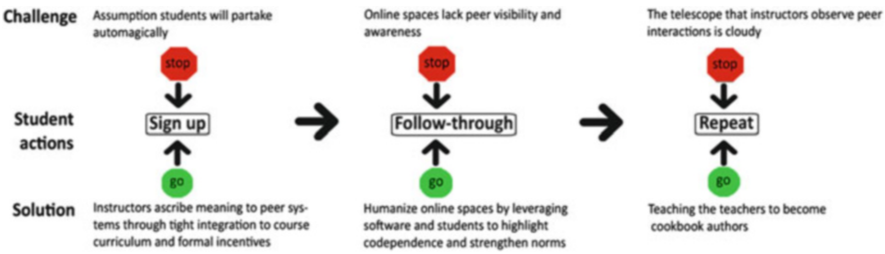


Fig. 11 The challenges and remedies of adoption of peer learning systems presented in this chapter

measures of efficacy include sign-up and follow-through rates, course participation and activity, and participation structure and duration. Qualitative data includes students' and instructors' comments in surveys and interviews. We describe how peer learning behavior varies with changing student practices, teacher practices, and course materials.

9 Social Capabilities Do Not Guarantee Social Use

Peer learning systems share many attributes with collaborative software more generally (Grudin 1994). However, the additional features of the educational setting change users' calculus. Throughout the deployments of our platforms, we've observed different approaches that instructors take when using our peer systems with their material.

Often, instructors dropped a platform into their class, then left it alone and assumed that students would populate it. For example, one course using Talkabout only mentioned it once in course announcements. Across four weeks, the sign-up rate was just 0.4 %, compared to a more successful sign-up rate of 6.6 % in another course; sign-up rate being the number of students who signed up to participate in the peer system out of the number of active students (students who watched a lecture video) in the course. Low percentages represent conservative estimates as the denominator represents students with minimal activity. When this theme recurred in other Talkabout courses, it was accompanied with the same outcome: social interactions languished. Why would instructors who put in significant effort developing discussion prompts introduce a peer learning system, but immediately abandon it?

Through discussions, we noticed that instructors assumed that a peer system would behave like an already-popular social networking service like Facebook where people come en masse at their own will. This point of view resonates with a common assumption that MOOC students are extremely self-motivated, and that such motivation shapes their behavior (Breslow et al. 2013; Kizilcec and Schneider 2015). In particular, instructors were *not* treating the systems like novel learning technology, but rather as bolted-on social technology. The assumption seemed to be that building a social space will cause students to just populate it and learn from each other.

However, peer learning systems may need more active integration. The value of educational experiences is not immediately apparent to students, and those that are worthwhile need to be signaled as important in order to achieve adoption.

Chat rooms underscored a similar point of the importance of pedagogical integration. Early chat room implementations were easily accessible (embedded in-page near video lectures) but had little pedagogical scaffolding (Coetzee et al. 2014). Later, more successful variants that strongly enforced a pedagogical structure were better received (Coetzee et al. 2015).

9.1 Peer Software as Learning Spaces

Even the best-designed peer learning activities have little value unless students overcome initial reluctance to use them. Course credit helps even students to commit, and those who have committed, to participate. Consider follow-though rates: the fraction of students who attend the discussion out of the students signed up for it. In an international women's rights course, before extra credit was offered, Talkabout follow-through rate was 31 %. After offering extra credit, follow-through rate increased to 52 %. In other classes, we've seen formal incentives raise follow-through rates up to 64 %.

Faculty can signal to students what matters by using scarce resources like grade composition and announcements. We hypothesize that these signals of academic importance and meaning increase student usage. For example, in a course where the instructors just repeatedly announced Talkabout in the beginning, 6.6 % of active students signed up, a large increase from the 0.4 % sign-up rate when there was only one mention of Talkabout.

We saw similar effects with PeerStudio. When participation comprises even a small fraction of a student's grade, usage increases substantially. In one class where PeerStudio was optional, the sign-up rate was 0.8 %. The fraction of users was six times higher in another class where use of PeerStudio contributed to their grade: the sign-up rate was 4.9 %. To maintain consistency with insights from Talkabout, sign-up rates for PeerStudio also represents the number of students who signed up to participate out of the number of active students (students in the course who watched a lecture video).

Students look up to their instructors, creating a unique opportunity to get and keep students involved. One indicator of student interest is if they visited the Talkabout website. Figure 12 shows Talkabout page views after instructors posted on the course site discussing Talkabout, and a decrease in page views when no announcement is made. Talkabout traffic was dwindling towards the end of the course, so the instructor decided to offer extra credit for the last round of Talkabout

Fig. 12 When instructors highlight peer learning software, students use it. Talkabout pageviews of a women's rights course. Instructor announcements are followed by the largest amount of Talkabout pageviews throughout the course. R1 represents Round 1 of Talkabout discussions, and so on, with orange rectangles framing the duration of each round. When instructor does not mention Round 4 and 6, pageviews are at their lowest

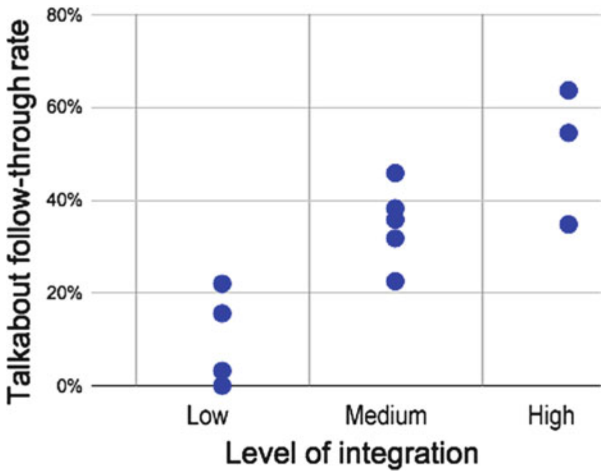


Fig. 13 Follow-through rate from 12 Talkabout courses increases as integration increases

discussions. During the extra-credit granting Talkabout discussions, page views increase around twofold the previous four rounds.

To understand how pedagogical integration and incentives, and follow-through rate interact, we divided 12 Talkabout courses into three categories, based on how well Talkabout was incentivized and integrated pedagogically (see Fig. 13). Courses that never mentioned Talkabout or mentioned it only at the start of the course are labeled "Low integration". Such courses considered Talkabout a primarily social opportunity, similar to a Facebook group. Few students signed up, and even fewer actually participated: the average follow-through rate was 10 %. The next category, "Medium integration," was well integrated but poorly incentivized, classes. These classes referred to Talkabout frequently in announcements, encouraged students to participate, and had well-structured discussion prompts, but they had no formal incentive. Such classes had an average follow-through rate of 35 %.

Well-incentivized and integrated classes, “High integration,” offered course extra credit for participation and continuously discussed Talkabout in course announcements, and averaged 50 % follow-through rate. This visualization highlights the pattern that the more integrated the peer learning platform is, the higher the follow-through rate is. We have found that offering even minimal course credit powerfully spurs initial participation, and that many interventions neglect to do this. As one student noted in a post-discussion survey, “I probably wouldn’t have done it [a Talkabout session] were it not for the five extra credit points but I found it very interesting and glad I did do it!”

The Talkabout course with the highest follow-through rate not only offered Talkabout for extra credit, but also offered technical support, including a course-specific Talkabout FAQ (Talkabout has an FAQ but it is not course specific). Looking at the forums, the role of the FAQ became apparent: many students posted questions about their technological difficulties and the community TAs and even other students would direct students to this FAQ—loaded with pictures and step by step instructions to help these students understand what Talkabout is and how it’s related to them. Moreover, the course support team answered any questions that could not be answered by the FAQ, ensuring that anyone who was interested in using the peer learning platform got the chance to do so.

However, online classes must also accommodate students with differing constraints from around the world. For instance, Talkabout is not available to some students whose country (like Iran) blocks access to Google Hangouts. Other students may simply lack sufficient reliable Internet bandwidth. One course offered small-group discussions for credit that were held either online (with Talkabout) or in-person in order to combat this challenge. When the strongest incentives are impractical, courses can still improve social visibility to encourage participation.

10 Social Transluscence Is Limited Online

Online students are “hungry for social interaction” (Kizilcec and Schneider 2015). Especially in early MOOCs, discussion forums featured self-introductions from around the world, and students banded together for in-person meet-ups. Yet, when peer-learning opportunities are provided, students don’t always participate in pro-social ways; they may neglect to review their peers’ work, or fail to attend a discussion session that they signed up for.

We asked 100 students who missed a Talkabout why they did so. 18 out of 31 responses said something else came up or they forgot. While many respondents apologized to us as the system designers, none mentioned how they may have let down their classmates who were counting on their participation. This observation suggests that social loafing may be endemic to large-scale social learning systems. If a student doesn’t feel responsible to a small set of colleagues and the instructor instead diffuses that responsibility across a massive set of peers, individuals will feel less compunction to follow through on social commitments.

To combat social loafing, we must reverse the diffusion of responsibility by transforming it onto a smaller human scale. Systems that highlight co-dependence may be more successful at encouraging pro-social behavior (Cialdini and Goldstein 2004). In a peer environment, students are dependent on each other to do their part for the system to work. Encouraging commitment and contribution can help students understand the importance of their participation, and create successful peer learning environments (Kraut et al. 2012).

10.1 Norm-Setting in Online Social Interaction

Norms have an enormous impact on people's behavior. In-person, teachers can act as strong role models and have institutional authority, leading to many opportunities to shape behavior and strengthen and set norms. Online, while these opportunities diminish with limited social visibility, other opportunities appear, such as shaping norms through system design. Platform designers, software and teachers can encourage peer empathy and mutually beneficial behavior by fostering pro-social norms.

Software can illuminate social norms online. For instance, when PeerStudio notices that a student has provided scores without written feedback, it reminds them of the reciprocal nature of the peer assessment process (see Fig. 14). As a different example, students that are late to a Talkabout discussion are told they won't be allowed to join the discussion, just as they'd not like to have a discussion interrupted by a late classmate. Instead, the system provides them an option to reschedule. Systems need not wait until things go wrong to set norms. From prior work, we know students are highly motivated when they feel that their contribution matters (Bransford et al. 2000; Ling and Beenen 2005). As an experiment, we emailed students in two separate Talkabout courses before their discussion saying that their peers were counting on them to show up to the discussion (see Fig. 15). Without a reminder email, only 21 % of students who signed up for a discussion slot actually showed up. With a reminder email, this follow-through rate increased to 62 %.

10.2 How Can We Leverage Software and Students to Highlight Codependence and Ascribe Meaning?

PeerStudio recruits reviewers by sending out emails to students. Initially, this email featured a generic request to review. As an experiment, we *humanized* the request by featuring the custom request a student had made. For example, the generic boilerplate request became the personalized request that the student had written

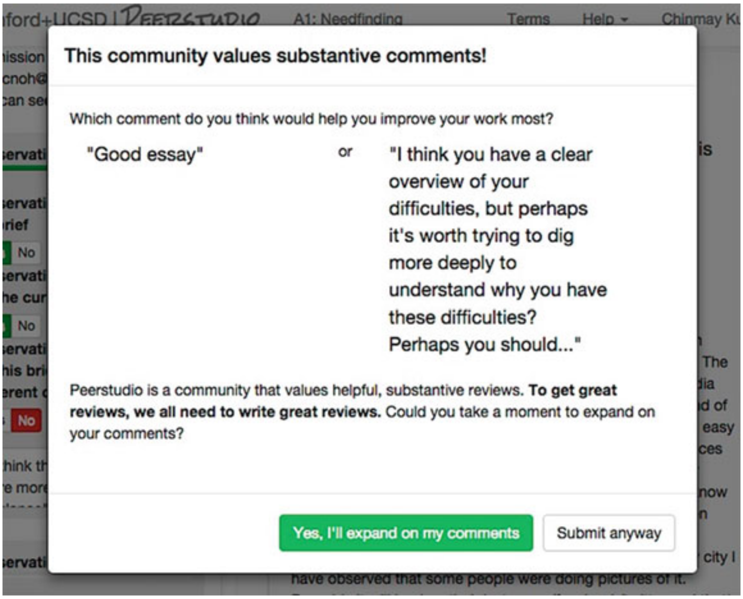


Fig. 14 When PeerStudio detects a review without comments, it asks the reviewer if they would like to go back and improve their review by adding comments

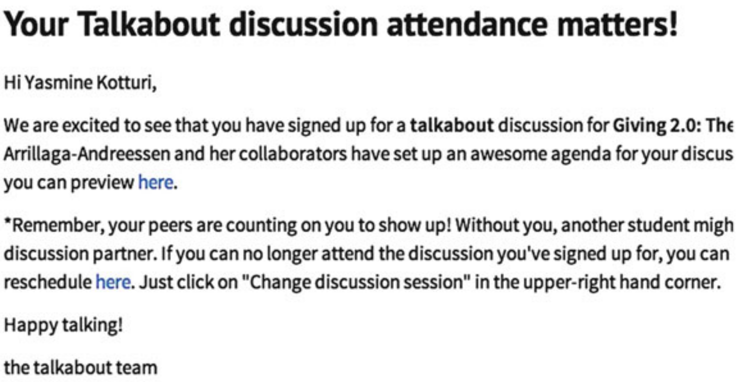


Fig. 15 An email sent to students prior to their discussion, reminding them of the importance of their attendance, increases follow-through rate 41 %

before submitting his draft. Immediately after making this change, review length increased from an average of 17 words to 24 words.

Humanized software is not the only influencer: forum posts from students sharing their peer learning experiences can help validate the system and encourage others to give it a try. For example, one student posted: “I can’t say how much I love discussions. . .and that’s why I have gone through 11-12 Talkabout sessions just to know, discuss and interact with people from all over the world.” Although unpredictable (Cheng et al. 2014), this word-of-mouth technique can be highly effective for increasing stickiness (Bakshy et al. 2009). When students shared Talkabout experiences in the course discussion forums (2000 posts out of 64,000 mentioned Talkabout, 3 %), the sign-up rate was 6 % (2037 students), and the follow-through rate was 63 %. However, the same course offered a year later, did not see similar student behavior (260 posts out of 80,000 mentioned Talkabout, 0.3 %). The sign-up rate was 5 % (930 students) and follow-through rate was 55 %. Although influenced by external factors, this suggests that social validation of the systems is important.

10.3 Leveraging Students' Desire to Connect Globally

Increasing social translucence has one final benefit: it allows students to act on their desire for persistent connections with their global classmates. For example, incorporating networking opportunities in the discussion agenda allocates times for students to mingle: "Spend 5 min taking turns introducing yourselves and discussing your background." However, we note that this is not a "one-size-fits-all" solution: certain course topics might inspire more socializing than others. For instance, in an international women's rights course, 93 % of students using Talkabout shared their contact information with each other (e.g. LinkedIn profiles, email addresses), but in a course on effective learning, only 18 % did.

11 Designing and Hosting Interaction from Afar

Like a cook watching a stew come to a boil and adjusting the temperature as needed, an instructor guiding peer interactions in-person can modulate her behavior in response to student reactions. Observing how students do in-class exercises and assimilating non-verbal cues (e.g., enthusiasm, boredom, confusion) helps teachers tailor their instruction, often even subconsciously (Klemmer et al. 2006).

By contrast, the indirection of teaching online causes multiple challenges for instructors. First, with rare exceptions (Chen 2001), online teachers can't see much about student behavior interactively. Second, because of the large-scale and asynchronous nature of most online classes, teachers can't directly coach peer interactions. To extend—and possibly butcher—the cooking metaphor, teaching online shifts the instructor from the in-the-kitchen chef to the cookbook author. Their recipes need to be sufficiently stand-alone and clear that students around the globe can cook up a delicious peer interaction themselves. However, most instructors lack the tools to write recipes that can be handed off and reused without any interactive guidance on the instructor's part.

11.1 Guidelines for Writing Recipes: Scaffolding Peer Interactions from Behind a Computer

Most early users of Talkabout provided both too little student motivation and discussion scaffolding. Consequently, usage was minimal (Kulkarni et al. 2015). Unstructured discussion did not increase students' academic achievement or sense of community (Coetzee et al. 2014). To succeed, we needed to specifically target opportunities for self-referencing, highlight viewpoint differences using boundary objects, and leverage students as mediators (Kulkarni et al. 2015). To understand this range of structure, we looked the discussions from 12 different courses and compared agenda word length and discussion duration. We split discussions into two categories: long and short discussion agendas, with 250 words as the threshold, and compared credit-granting and no credit discussions (see Fig. 16). Average discussion duration was 31 min for short agendas. However, only those long agendas that awarded credit successfully incentivized students to discuss longer: the average discussion with credit was 49 min, and without was 30 min. All agendas asked students to discuss for 30 min; students were staying the extra time voluntarily.

We worried that over-structuring an interaction with lengthy and tiresome agendas would leave no space for informal bond-building. However, even with sufficient structure, students can easily veer from the schedule and socialize, exchange social networking information, and offer career advice.

Software systems, platforms, and data-driven suggestions each play a more active role in helping teachers create effective recipes. While most early Talkabout instructors provided too little discussion scaffolding, our data showed instincts led early Talkabout instructors to worry too much about scheduling. For example, time

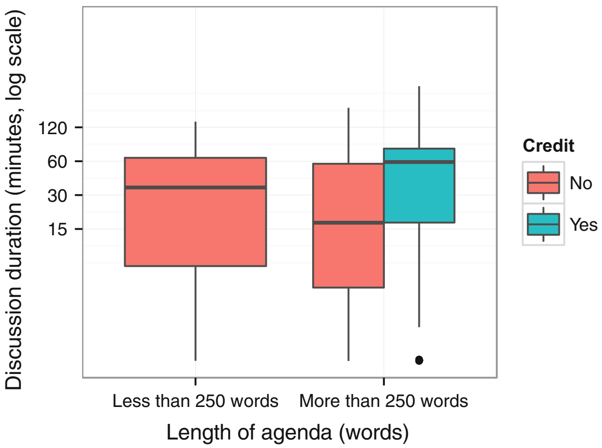


Fig. 16 Longer discussion agendas incentivize students to discuss longer, but only when they are accompanied by course credit for participation

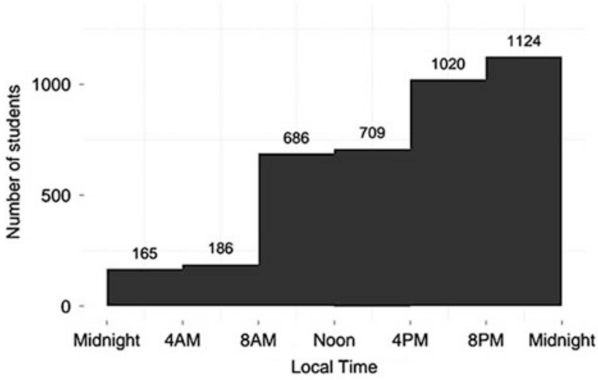


Fig. 17 Data from nine classes and 3400 students shows that most students discuss in the evening, but there are students that will discuss at all 24 h

zones are a recurring thorn in the side of many types of global collaboration, and peer learning is no exception. Every Talkabout instructor was concerned about discussion session times and frequency, as this a major issue with in-person sections. Instructors often asked if particular times were good for students around the world. Some debated: would 9 pm Eastern Time be better than 8 pm Eastern Time, as more students would have finished dinner? Or would it be worse for students elsewhere? Other instructors were unsure of how many discussions timeslots to offer. One instructor offered a timeslot every hour for 24 h because she wanted to ensure that there were enough scheduling options. However, an unforeseen consequence of this was that the participants were too spread out over the 24 discussions, and thus some students were left alone.

Analyzing when students participate in discussions taught us that most students prefer evenings for discussions. Yet, different students prefer different times, with every time of day being preferred by someone (Fig. 17). This data suggests that it is unimportant for instructors to find a particular scheduling "sweet spot," and instead their time is better utilized elsewhere: creating the discussion agendas, for example. In summary, these examples illustrated where intuitions can lead teachers and system designers astray. Data-driven suggestions are important to transform expert cooks into cookbook authors.

12 Teaching Teachers by Example

Even fantastic pedagogical innovation can be hamstrung when there is a mismatch between curricular materials and platform functionality. When curricula did not match to the needs of the setting, the learning platforms languished. We emphasize the importance of *teaching by example*: creating designs and introductory experiences that nudge instructors toward the right intuitions. While always true with educational innovation, the online education revolution is a particularly dramatic change of setting, and instructor scaffolding is particularly important.

One of the most robust techniques we have found for guiding instructors is to provide successful examples of how other teachers have used the learning platform. In many domains, from design to writing research papers, a common and effective strategy for creating new work is to template off similar work that has a related goal (Klemmer 2015). During interviews with Talkabout instructors, a common situation recurred: the instructor was having a hard time conceptualizing the student experience. Therefore, to help instructors navigate the interface and create effective discussion prompts, we added an annotated example of a Talkabout discussion (see Fig. 1). Still, we observed that many instructors had difficulty creating effective discussion agendas, e.g. they were very short and did not leverage the geographic diversity Talkabout discussions offer. As an experiment, we walked an instructor through Talkabout—in a Talkabout—and showed an excellent example agenda from another class. This helped onboard the new instructor to working with Talkabout: she was able to use the example as a framework that she could fill in with her own content (see Fig. 18). Next, we showed example course

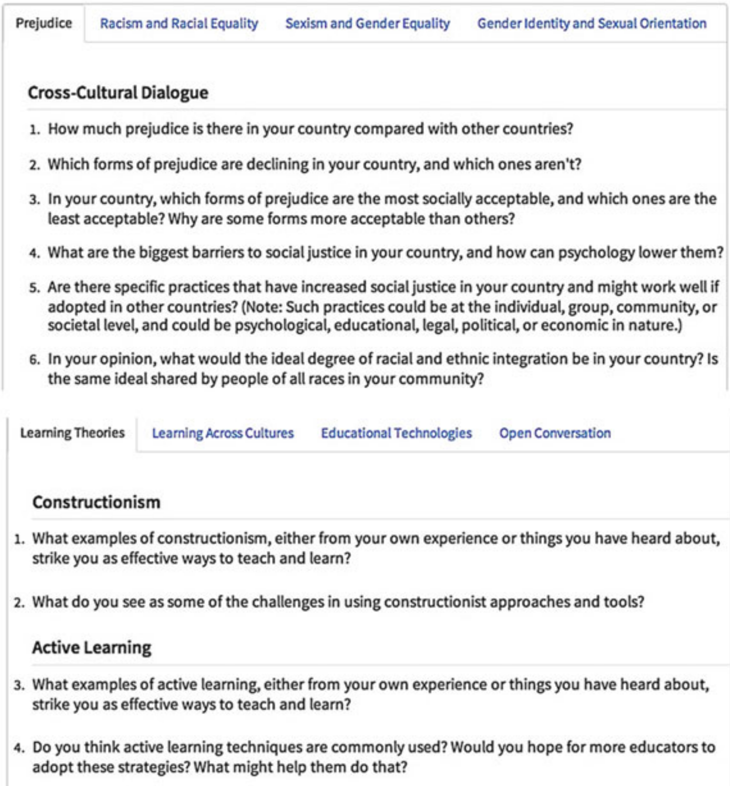


Fig. 18 Two discussions prompts: top used as a template to show a new Talkabout instructor an excellent example. Bottom prompt generated based off the example

announcements that described Talkabout using layman's terms and offering pictures of the Talkabout discussion. Since course announcements are viewed by most online students, it is important to describe peer learning platforms in basic terms to convey a straight forward message.

The next step was to help instructors gain an understanding of what occurs during student discussions. To do this, we showed an instructor a video clip of a Talkabout discussion along with a full discussion summary. In response, the instructor said, "The most interesting point was around the amount of time each student spoke. In this case, one student spoke for more than half of the Talkabout. This informs us to be more explicit with time allocations for questions and that we should emphasize that we want students to more evenly speak." By helping her visualize the interactions, she was able to restructure her discussion prompts in order to achieve her desired discussion goal; in this case, encouraging all students to have equally share their thoughts.

13 Conclusion

This chapter first suggests that the scale of massive online classes enables systems that drastically and reliably reduce the time to obtain feedback and creates a path to iteration, mastery and expertise. These advantages can also be scaled-down to in-person classrooms. In contrast to today's learn-and-submit model of online education, we believe that the continuous presence of peers holds the promise of a far more dynamic and iterative learning process. This chapter then provides evidence for three challenges to global-scale adoption of peer learning, and offered three corresponding socio-technical remedies. We reflect on our experience from developing, designing and deploying our social learning platforms: Talkabout and PeerStudio, as well as our experience as teachers in physical and online classes. We looked at student practices, teacher practices and material design, and assessed the relationship between those and peer learning adoption. When peer systems and curricula are well integrated, the social context is illuminated, and teachers' and system designers' intuitions for scaffolding are guided by software, students do adopt these systems.

Acknowledgments We are grateful to Scott Plous, Ed Hutchins and all instructors who used Talkabout and PeerStudio for their thought-provoking discussions and feedback; Courtney Noh and Julia Cambre for contributing to software development; Coursera and OpenEdX for platform integration and encouraging instructors to try PeerStudio; and the many students who taught and learned with their peers and shared their stories with us. This research was funded in part through NSF grants #1351131 and #1444865, the Hasso Plattner Design Thinking Program, and the Siebel Scholars Program. This research was conducted under Stanford IRB protocol #30324 and #30319.

References

Anderson S, Rodin J (1989) Is bad news always bad?: cue and feedback effects on intrinsic motivation. J Appl Soc Psychol 19(6):449–467

Andrade HG (2001) The effects of instructional rubrics on learning to write. Current Issues Educ 4:4

Andrade HG (2005) Teaching with rubrics: the good, the bad, and the ugly. Coll Teach 53(1): 27–31

André P, Bernstein M, Luther K (2012) Who gives a tweet? In: Proceedings of the ACM 2012 conference on computer supported cooperative work—CSCW'12, ACM Press, p 471

Bakshy E, Karrer B, Adamic LA (2009) Social influence and the diffusion of user-created content. In: Proceedings of the 10th ACM conference on electronic commerce, pp 325–334

Balcazar FE, Hopkins BL, Suarez Y (1986) A critical, objective review of performance feedback. J Organ Behav Manag 7:2

Bransford J, Brown A, Cocking R (2000) How people learn. The National Academies Press, Washington, DC

Bransford J, Schwartz DL (1999) Rethinking transfer: a simple proposal with multiple implications. Rev Res Educ 24:61–100

Breslow L, Pritchard D, DeBoer J, Stump G, Ho A, Seaton D (2013) Studying learning in the worldwide classroom: Research into edX's first MOOC

Brooks M, Basu S, Jacobs C, Vanderwende L (2014) Divide and correct: using clusters to grade short answers at scale. Learning at Scale

Buxton B (2007) Sketching user experiences: getting the design right and the right design. Morgan Kaufmann, San Francisco

Cambre J, Kulkarni C, Bernstein MS, Klemmer SR (2014) Talkabout: small-group discussions in massive global classes. Learning@Scale

Carlson PA, Berry FC (2003) Calibrated peer review and assessing learning outcomes. In: Frontiers in education conference

Chen M (2001) Design of a virtual auditorium. In: MULTIMEDIA'01 Proceedings of the ninth ACM international conference on multimedia, pp 19–28

Cheng J, Adamic L, Dow P, Kleinberg J, Leskovec J (2014) Can cascades be predicted? In: Proceedings of the 23rd international conference on world wide web, pp 925–936

Cialdini R, Goldstein N (2004) Social influence: compliance and conformity. Annu Rev Psychol 55:591–621

Coetzee D, Fox A, Hearst MA, Hartmann B (2014) Chatrooms in MOOCs: all talk and no action. In: Proceeding of the ACM conference on learning @ scale, ACM Press, pp 127–136

Coetzee D, Lim S, Fox A, Hartmann B, Hearst MA (2015) Structuring interactions for large-scale synchronous peer learning. In: CSCW: ACM Conference on computer supported collaborative work

Crouch CH, Mazur E (2001) Peer instruction: ten years of experience and results. Am J Phys 69(9):970

Dawes RM (1979) The robust beauty of improper linear models in decision making. Am Psychol 34(7):571

Dourish P, Bell G (2007) The infrastructure of experience and the experience of infrastructure: meaning and structure in everyday encounters with space. Environ Plan B Plan Des 34(3):414

Dourish P, Bellotti V (1992) Awareness and coordination in shared workspaces. In: Proceeding CSCW'92 Proceedings of the 1992 ACM conference on computer-supported cooperative work, pp 107–114

Dow SP, Heddleston K, Klemmer SR (2009) The efficacy of prototyping under time constraints. In: Proceeding of the ACM conference on creativity and cognition, ACM Press, p 165

Dow S, Fortuna J, Schwartz D, Altringer B, Klemmer S (2011) Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In: Proceedings of the 2011 annual conference on human factors in computing systems, pp 2807–2816

Erickson T, Kellogg W (2000) Social translucence: an approach to designing systems that support social processes. In: ACM Transactions on Computer-Human Interaction (TOCHI)—Special issue on human-computer interaction in the new millennium, Part 1 7(1), pp 59–83
Ericsson KA, Krampe RT, Tesch-Römer C (1993) The role of deliberate practice in the acquisition of expert performance. Psychol Rev 100:3
Falchikov N, Goldfinch J (2000) Student peer assessment in higher education: a meta-analysis comparing peer and teacher marks. Rev Educ Res 70(3):287–322
Fast E, Lee C, Aiken A, Bernstein MS, Koller D, Smith E (2013) Crowd-scale interactive formal reasoning and analytics. In: Proceedings of the 26th annual ACM symposium on user interface software and technology
Gray WD, Boehm-Davis DA (2000) Milliseconds matter: an introduction to microstrategies and to their use in describing and predicting interactive behavior. J Exp Psychol 6(4):322–335
Greenberg S (2009) Embedding a design studio course in a conventional computer science program. In: Creativity and HCI: From Experience to Design in Education. Springer, pp 23–41
Grudin J (1994) Groupware and social dynamics: eight challenges for developers. Commun ACM 37(1):92–105
Guskey TR (2007) Closing achievement gaps: revisiting Benjamin S. Bloom's "Learning for Mastery". J Adv Acad 19(1):8–31
Hattie J, Timperley H (2007) The power of feedback. Rev Educ Res 77(1):81–112
Heffernan N, Heffernan C, Dietz K, Soffer D, Pellegrino JW, Goldman SR, Dailey M (2012) Improving mathematical learning outcomes through automatic reassessment and relearning. AERA
Kim S-M, Pantel P, Chklovski T, Pennacchiotti M (2006) Automatically assessing review helpfulness. In: Proceedings of the 2006 conference on empirical methods in natural language processing, association for computational linguistics, pp 423–430
Kittur A, Chi EH, Suh B (2008) Crowdsourcing user studies with mechanical turk. In: Proceeding of CHI, ACM Press, p 453
Kizilcec RF, Schneider E (2015) Motivation as a lens to understand online learners: towards data-driven design with the OLEI scale
Klemmer SR (2015) Katayanagi Lecture at CMU: the power of examples
Klemmer SR, Hartmann B, Takayama L (2006) How bodies matter: five themes for interaction design. In: Proceedings of the 6th conference on designing interactive systems, pp 140–149
Kluger AN, DeNisi A (1996) The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. Psychol Bull 119(2): 254–284
Konstan JA, Walker JD, Brooks DC, Brown K, Ekstrand MD (2014) Teaching recommender systems at large scale. In: Proceeding of the ACM conference on learning @ scale conference, ACM Press
Kraut RE, Resnick P, Kiesler S, Burke M, Chen Y, Kittur N, Konstan J, Ren Y, Riedl J (2012) Building successful online communities: evidence-based social design (Google eBook), MIT Press
Krosnick JA (1999) Survey research. Annu Rev Psychol 50(1):537–567
Kulik JA, Kulik C-LC (1987) Timing of feedback and verbal learning. Rev Educ Res 58(1):79–97
Kulkarni C, Wei KP, Le H, Chia D, Papadopoulos K, Cheng J, Koller D, Klemmer SR (2013) Peer and self assessment in massive online classes. ACM Trans Comput-Hum Interact (TOCHI) 20(6):33
Kulkarni C, Socher R, Bernstein MS, Klemmer SR (2014) Scaling short-answer grading by combining peer assessment with algorithmic scoring. In: ACM conference on learning @ scale, ACM
Kulkarni C, Cambre J, Kotturi Y, Bernstein M, Klemmer S (2015) Talkabout: making distance matter with small groups in massive classes. In: CSCW: ACM conference on computer supported collaborative work
Latham GP, Locke EA (1991) Self-regulation through goal setting. Organ Behav Hum Decis Process 50(2):212–247

Ling K, Beenen G (2005) Using social psychology to motivate contributions to online communities. J Comput-Mediat Commun 10(4):00
Marsh RL, Landau JD, Hicks JL (1996) How examples may (and may not) constrain creativity. Mem Cogn 24(5):669–680
O'Donnell AM, Dansereau DF (1995) Scripted cooperation in student dyads: a method for analyzing and enhancing academic learning and performance. In: Interaction in cooperative groups: the theoretical anatomy of group learning. Cambridge University Press, Cambridge, UK, pp 120–141
Plattner H, Meinel C, Leifer L (eds) (2015) Design thinking research: making design thinking foundational. Springer, Heidelberg
Porter L, Lee CB, Simon B (2013) Halving fail rates using peer instruction: a study of four computer science courses. In: Proceeding SIGCSE'13 Proceeding of the 44th ACM technical symposium on computer science education, pp 177–182
Rosenberg J, Lorenzo M, Mazur E (2007) Peer instruction: making science engaging. In: Handbook of college science teaching. NSTA Press, Arlington, VA, pp 77–85
Smith M, Wood W, Adams W, Wieman C, Knight J, Guild N, Su T (2009) Why peer discussion improves student performance on in-class concept questions. Science 323(5910):122–124
Sommers N (1982) Responding to student writing. Coll Compos Commun 33(2):148–156
Stephens-Martinez K, Hearst MA, Fox A (2014) Monitoring MOOCs: which information sources do instructors value? In: Proceedings of the first ACM conference on learning @ scale conference, pp 79–88
Tinapple D, Olson L, Sadauskas J (2013) CritViz: web-based software supporting peer critique in large creative classrooms. Bull IEEE Techn Comm Learn Technol 15(1):29
Turkle S (2011) Alone together: why we expect more from technology and less from each other. Basic Books, New York

Developing Instrumentation for Design Thinking Team Performance

Neeraj Sonalkar, Ade Mabogunje, Halsey Hoster, and Bernard Roth

Abstract Multidisciplinary teamwork is a key requirement in the design thinking approach to innovation. Previous research has shown that team coaching is an effective way to improve team performance. However, the tools currently available for effective team coaching are limited to heuristics derived from either experienced design thinking professionals or clinical psychology practitioners. Our research aims to improve this situation by providing design thinking managers, coaches, and instructors a reliable instrument for measuring design team performance. In this chapter, we present the underlying methodology for instrument design. The development of a specific diagnostic instrument, based on a visual notation called the Interaction Dynamics Notation, is explained in terms of both the workflow of data through the instrument and the exploratory studies conducted to design the instrument user interface.

1 Introduction

Organizations are increasingly adopting design thinking as an approach for promoting product, process, and service innovation. In the past decade the term *design thinking* has developed multiple interpretations, including a process for innovation (Brown 2008), an approach towards problem solving (Dorst 2011), a personal creative mindset (Rauth et al. 2010), and an organizational culture oriented towards innovative output (Kolko 2015). Rapid and enthusiastic adoption of design thinking has created an urgent need to deepen our scientific understanding of design thinking for two reasons: first, to prevent shallow interpretations that propagate design thinking as a business fad, and second, to enable organizations to reach peak performance in their design thinking practice. In this chapter, we present research

N. Sonalkar (✉) • A. Mabogunje • H. Hoster
Center for Design Research, Stanford University, Stanford, CA, USA
e-mail: sonalkar@stanford.edu

B. Roth
Hasso Plattner Institute of Design, Stanford University, Stanford, CA, USA

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_15

to develop an instrument that deepens our understanding of design teams. We developed and refined this instrument with the intention to improve teams' design thinking performance.

2 Why Instrument Design Thinking Teams?

Instrumentation is defined as the development and use of devices that reliably measure a phenomenon of interest. Instruments provide methods to both sense and represent a phenomenon within a measurement scheme; these methods go beyond the limitation of language usage and the subjectivity of individual observers (Baird 2004). Reliable, repeatable, and measurable understanding of a phenomenon is central to the development of scientific knowledge, and subsequently to our ability to engineer and improve the phenomenon.

Considering design thinking broadly as an approach to problem solving that has been adapted from *how designers think and act*, research in design thinking is rooted in the long history of design theory and methodology research in various design disciplines like architecture, engineering design, and product design (Johansson-Sköldberg et al. 2013). However, much of this research is descriptive in nature and lacks instrumentation that could both improve rigor and render the research accessible to practitioners. In this chapter, we present the early stages in the development of one such instrument for design thinking team interactions.

As design thinking matures, we could imagine instruments that measure concept generation, prototyping, or framing that can give teams an understanding of their design activity that goes beyond the limitations of human sensing and allows for the development of high performance design thinking teams that can innovate faster, more reliably, and more efficiently.

3 How to Develop an Instrument for Design Thinking Teams?

Developing a measuring instrument involves sensing the phenomenon of interest and then comparing it with a standard measurement scheme to give an output in units of the agreed standard. The problem with developing an instrument for design thinking is twofold:

1. Design thinking is a complex socio-technical activity that is largely dependent on human interaction and language, which makes sensing the phenomenon inherently subjective.
2. There are no agreed upon standards for specific parameters of design thinking.

Developing an instrument for design thinking involves both designing a sensing device and developing a measuring standard for the phenomenon being measured. The development of a measuring standard would require numerous studies of the design thinking parameter being measured over a number of different contexts to map out a spectrum of variation and determine benchmarks that could be consistent across the different instances. This substantial research effort would benefit from first developing a sensing device that could represent the design parameter being studied in a manner that facilitates analysis. Thus, the first step towards developing a measuring instrument for design thinking is to develop a sensing device. In this chapter, we present the development of such a sensing device for measuring the development of concepts, decisions, or frames through interpersonal interactions in a design team.

4 Interaction Dynamics Notation as a Design Instrument

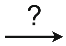
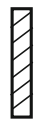
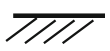
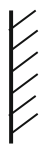
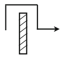
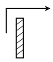
The Interaction Dynamics Notation (IDN) is a visual notation that models interpersonal interactions in the context of a design activity, such as concept generation, framing of user need, or design reviews (Sonalkar et al. 2013). The relevant parameters that we want to analyze being developed through such design interactions can be highlighted in the IDN output.

The theoretical foundations of IDN are based on cognitive semiotics and theatrical improvisation. The domain of cognitive semiotics employs a notation called as the Force Dynamics Notation (Talmy 1988) that visualizes the meaning conveyed by phrases such as "she let him down" (Brandt 2004, page 42). The Force Dynamics Notation visualizes the meaning in a phrase as a narrative plot, where actors experience forces like "letting down" that affect their narrative journey. Interactive Dynamics Notation, or IDN, adapts this notation for design interactions. In its current implementation, IDN captures the meaning conveyed in a speaker turn—verbal or non-verbal—in terms of the force it has on the subsequent response. For example, we say designer A asked a question, not because of the intonation in her tone, or her inferred intention, but because a team member B responded to A as if answering a question. If no one responds to A, her turn is not coded as a question.

IDN consists of 12 symbols. The original IDN set of symbols, published earlier, has evolved over time through new experiments and further research. They now include ignored and ambiguity. Deviation and interruption are no longer in the symbol set. The current set of IDN symbols are outlined in Table 1.

Consider an instrument based on the Interaction Dynamics Notation. Let's call it an IDN instrument. The IDN instrument takes a conversation from video data and converts it into a visual representation of a sequence of symbols. Further, relevant design parameters are highlighted in the representation to analyze design relevant constructs emerging through conversation. For example, in concept generation activity, the responses can be highlighted to map out how concepts are co-created through interpersonal interactions. Figure 1 shows the operations that constitute the

Table 1 The symbols that comprise the interaction dynamics notation

Move  A A “move” indicates that a speaker has made an expression that moves the interaction forward in a given direction.	Question  A A question indicates an expression that elicits a move.	Silence  Silence is a state in the conversation when none of the participants speak as they are engaged in other individual level activities.	Block  C Block indicates an obstruction to the content of the previous move.
Support for move  B Support-for-move indicates that the speaker agrees with and supports the previous move.	Support for block  C Support indicates an acceptance of a block by another person.	Overcoming B  C Overcoming a block indicates that though a block was placed in front of a move, a speaker was able to overcome the block and persist on course of the original move.	Deflection A  B When a speaker blocks a previous speaker’s move, that speaker or another can deflect the block with a move that presents an alternative direction for the interaction.
Yes and  C A move is considered to be a “Yes and” to the previous move if it accepts the content of the previous move and adds on to it.	Humor  A,B Humor indicates instances of shared laughter in teams.	Ignored  A Ignored implies that speaker A’s utterance was actively ignored by others on the team.	Ambiguous  X Ambiguous denotes a researcher’s inability to assign a symbol to that speaker utterance due to improper audio/video or indistinguishable speech.

IDN instrument. The instrument takes video data as input and, as output, gives the sequences and times of occurrence (or event time) of the 12 IDN symbolized interaction responses. The instrument achieves this conversion by first identifying speaker turns in video data. Then multiple analysts code the speaker identified video data with IDN *individually*. Their individual outputs are checked for data reliability and inter-analyst agreement. If found to be reliable and containing a relatively high level of agreement between analysts, the analyst output is considered as the instrument output. If the analysts’ output is not found to be reliable, analysts collectively code the video as a team to identify and resolve disagreements.

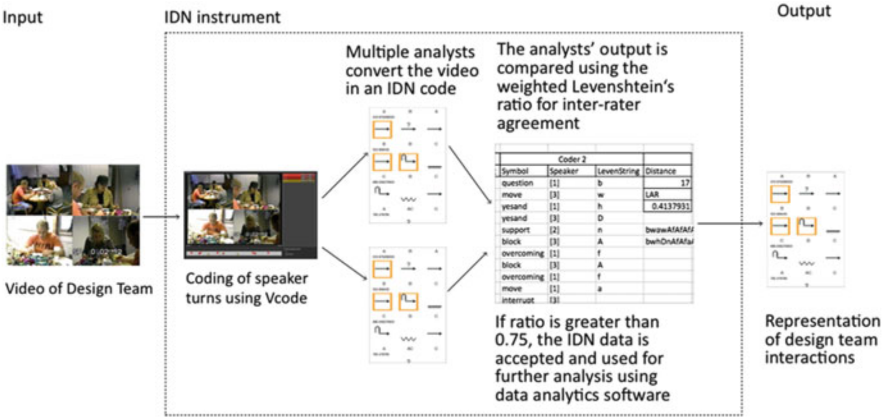


Fig. 1 The operations that constitute the IDN instrument

5 Stages of Development for the IDN Design Instrument

Transforming IDN from a visual notation to an instrument involved developing a reliable and efficient process for converting video of team interactions into an abstract representation of interaction patterns. As of today, assigning IDN symbols to speaker turns in a design team video cannot be accomplished computationally because of the variability of human conversation and the need to analyze participant responses in relation to each other. The coding of video into IDN symbols requires human judgment. Hence, a reliable and efficient human analyst is essential in the development of a process to convert design team video into IDN representation. The following processes were developed to hire and train undergraduate students into reliable and efficient IDN analysts.

5.1 Hiring and Training Analysts

5.1.1 Hiring of IDN Analysts

The guiding questions for developing a hiring process for people to become IDN analysts were:

1. Does a person need to have specific aptitude to be an IDN analyst?
2. How might we test the aptitude of candidates for IDN analyst positions?

We took the Emotion Coding Lab[1] established by Janine Giese-Davis at the Stanford Psychiatry Department as a role model. The Emotion coding lab trained

[1] http://stressandhealth.stanford.edu/people/giese-davis.html

students to code video data with specific coding schemes; students categorized the moment-to-moment emotion responses of the participants in the video. The lab screened applicants through an interview process involving video observation. Applicants who could detect emotions of people in their lay description of events occurring in the video were hired as coders, and applicants who tended to focus on words and did not notice emotions were not hired for coding emotions. Considering the coding of the Interaction Dynamics Notation, we identified the following aptitude parameters as prerequisites for becoming an IDN analyst:

1. Motivation to study innovation and design thinking
2. Curiosity to understand teamwork
3. Ability to concentrate on video data for long intervals of time
4. Ability to detect resistance to ideas expressed in the video. IDN codifies this resistance as a block.
5. Ability to identify "building on" of ideas. IDN codifies this as yes-and.
6. Sensitivity to non-verbal gestures and movements.

These parameters were tested in an interview process, which included asking candidates to describe what they noticed in specific video clips by highlighting interactions with blocks, yes-and, and non-verbal gesturing. The candidates who could correctly identify these interactions, and who showed motivation to study teamwork and design thinking, were selected for IDN analyst positions.

5.1.2 Training of IDN Analysts

The candidates who joined as IDN analysts participated in an 8-h training program followed by 2 weeks of individual practice on video clips of varying levels of complexity. Interspersed in the training were consensus coding sessions where the entire group, including the research team, coded a video together. Table 2 describes the activities conducted and the rationale for including them in the training program.

After their training, the analysts were given video data to code with IDN. They submitted the coded video files and IDN outputs files back to the research team, which were compared with the codes from different analysts to ensure data reliability.

5.2 Improving Reliability of Coding

Since the conversion of video in an IDN representation required human analysts to code the video, there was a need to design ways to ensure that inter-coder reliability so that the instrument output does not vary from one analyst to another. The following reliability measures were developed over multiple iterations of using and refining the IDN instrument.

Table 2 IDN Training program elements

Activity	Description	Rationale
Experiencing interactions	Participants engage in improve games such as sound ball, "I am a tree," "Yes Let's," and word-at-a-time story. Participants also practice concept generation with response constraints such as saying no, yes-but and yes-and.	These activities enable IDN trainees to have an experiential understanding of the concepts that we use in IDN.
Using interaction metaphors to describe design conversation	Participants are exposed to the role of metaphors in concept formation. They are asked to draw a metaphor for design conversations that they have participated in the past, and then share with the group.	Following Lakoff and Johnson (2008), metaphors are key to how we form concepts. Asking IDN trainees to engage in developing their own metaphors for conversations helps them conceptualize from their own experience. This activity in turn sensitizes them to how IDN was developed as a notational scheme.
Introduction to IDN	Participants learn how and why the Interaction Dynamics Notation was developed and what the different symbols of IDN are. They are also introduced to the IDN Tool.	This session is a formal introduction to IDN as a visual notation.
Consensus coding	IDN trainees and instructors code video clips together using the Interaction Dynamics Notation.	Consensus coding helps trainees learn how to assign IDN symbols in a social setting that facilitates peer learning.
Individual practice	IDN trainees code video with the IDN Tool and then check their work against the IDN output previously coded by the instructor. The video clips are of varying levels of difficulty. The trainees do not advance to next level until their earlier level output matches the benchmark.	Individual practice helps trainees in building expertise.
Professional vision session	Once the trainees have practiced IDN for 2–3 weeks, they discuss what makes them effective as IDN analysts. They share what they look for in video data, how they deal with maintaining attention on video, and describe their biases and difficulties as IDN analysts.	Describing how they code video as IDN analysts makes them aware of what goes into being a professional IDN analyst. The professional vision, which encapsulates how analyst should behave, is shared amongst the group of analysts.

5.2.1 Developing a Data Reliability Metric

We approached the task of finding a suitable inter-observer agreement metric as a design problem, starting with the design requirements. We identified the following requirements:

1. The metric needs to calculate agreement at the sequence level instead of at each individual position level.
2. The metric needs to be based on known statistical measures with proven usage.
3. The metric needs to be easy to use.

With these requirements in mind, we evaluated the inter-observer agreement metrics commonly used in psychology and education research that routinely uses human coders. Researchers currently use a number of different inter-observer agreement metrics to compare the data of human analysts. These range from simple percentage agreement to more sophisticated metrics such as Cohen's Kappa (Cohen 1960; Hubert 1977), which takes into account chance agreement. However, on detailed evaluation, these metrics did not meet the criteria of calculating agreement at the sequence level instead of at individual position level. For example, metrics like Cohen's Kappa measure agreement at the level of each individual position, as explained as follows.

If we have a pair of sequences "abcdeabcdeabcde" and "bcdeabcdeabcdea," Cohen's Kappa will give perfect disagreement since it will detect that each place in the first sequence does not match with the corresponding place in the second sequence. However, if we analyze the pair at a sequence level, we see that sequence 2 is an offset of sequence 1 by just the letter "a," which has been moved from the beginning of the sequence to the end of it. Since IDN notation needs to be compared at a sequence level and not at the level of each position in the sequence, we needed a different metric.

On conducting further literature review to study domains that deal with sequence level comparisons, two disciplines emerged as promising: computational linguistics, which involves alphabetic string comparisons such as spell checkers, and bioinformatics, which matches genetic sequences. Both fields used a basic algorithm to calculate the Levenshtein's Distance or Edit Distance between two sequences: the number of insertions, deletions or substitutions required to convert one sequence into another (Yujian and Bo 2007). For example, it takes 2 substitutions and 1 deletion to convert the sequence SITTING into KITTEN (substitute K for S, E for I, and delete G). Hence, the Levenshtein's distance between SITTING and KITTEN is 3. We found that Levenshtein's distance meets our criteria for measuring inter-observer agreement for IDN. Since IDN is a string of symbols, it can be easily converted into an alphanumeric form and then two sequences can be compared using the standard Levenshtein's distance algorithm. To further normalize the metric, we developed the Levenshtein's Agreement Ratio (LAR) which is defined as (length of longest sequence—Levenshtein's distance between them)/ length of the longest sequence. The LAR value for the above example of SITTING and KITTEN is 0.57. The value of LAR varies between 1 for perfect agreement to 0 for perfect disagreement. After evaluating the acceptable values for various other metrics, a minimum LAR value of 0.7 was considered acceptable for IDN data.

A custom defined function was created in Microsoft Excel to calculate LAR for IDN sequences. However, on further testing with actual IDN data coded by analysts, we realized that a weighted Levenshtein's distance (Ziółko et al. 2010) could be used since some symbol mismatches do not have much impact on further analysis. For example, missing an interruption symbol is of lower consequence than missing a block. Hence, a weighted Levenshtein's Agreement Ratio (wLAR) was developed by considering a lower penalty for disagreements regarding missing interruption, substituting block with overcoming, block with block-support, deflection with overcoming, yes-and with yes-and-question and support with yes-and. The same acceptable value of at least 0.7 was retained for wLAR.

5.2.2 Refining IDN Symbol Set and Rules

The wLAR range for acceptable IDN data was between 0.7 and 1. However, the analysts frequently produced IDN output that measured between 0.4 and 0.6. On examination of the IDN sequences that varied beyond the acceptable range, we observed that analysts often disagreed in assigning speaker turns, and in coding of specific symbols such as ‘support’ and ‘deviation’. Hence, we re-evaluated the rules of the IDN symbol set to better model design team interactions, and improve the reliability inherent in the rules of the symbols. The rules of some of the symbols such as ‘support’ were modified to remove elements that influenced disagreements. The symbol ‘interruption’ was eliminated, as it did not have an as much impact on the final model as expected. We added symbols ‘ignored’ and ‘ambiguous’ to model conversations more accurately. This refinement of the IDN symbol set and rules helped increase the wLAR value to 0.7–0.75.

6 Developing Requirements for IDN Instrument User Interface

Our objective in developing the IDN instrument was not just to measure design thinking team interactions, but also to influence their interactions to improve performance. The IDN instrument thus required a user interface that could provide designers with readouts that are relevant and useful to their team performance. In this section, we describe three studies performed with the IDN instrument to understand what kind of instrument feedback design teams find useful and relevant. The studies were designed to vary on three different dimensions: in-situ versus in-lab recording, delayed versus real-time feedback, and human-mediated versus direct feedback.

6.1 Study 1: Lab-Based Human Mediated Delayed Feedback

This study was conducted in collaboration with a graduate level course on creativity and innovation at the Hasso Plattner Institute for Design (d. School) at Stanford University. The instructors had given a 3-week project to seven student teams. The researchers worked with these seven teams to record a video of their on-going design conversations in a lab setting, and then gave them an IDN report of their team interactions within a period of 5 days. We recorded a 30-min video segment of team activity, and analyzed 15 min of the video segment using IDN instrument. From each team's IDN data, a few significant response types were identified and codified in a report format. Figure 2 shows the IDN feedback report of one of the teams, which were handed out to each team in a class session. The researchers explained their analysis for about 10 min and enacted a few interactions. Thereafter, teams had 10 min to discuss the reports with their team members. The reports contained some observations and guiding questions for discussing these observations within the team. The class session ended with an open Q&A session with the researchers. As per their class practice, the students submitted written reflections on the feedback session, which the instructors shared with the research team.

The session reflections showed the impact the feedback session had on some of the students. Below are few of the reflections noted by the students.

<<Team photo with A, B, C identifiers>>

Out of the 30 min interaction session, 15 min of video were analyzed using the Interaction Dynamics Notation.

A. Response Types Occurring in Interaction

	A	B	C	Total
Move	15	17	30	62
Question	3	17	6	26
Support	25	16	18	59
Humor	6	5	6	17
Yes And	3	10	10	23
Block	1	1	2	4
Overcoming	0	0	0	0
Deflection	1	2	0	3

Your team showed a large amount of humor i.e. your team had the maximum instances of laughter among all the teams analyzed. This behavior, along with support, exhibits a high level of group cohesion. Your group also asked a large amount of questions. Asking questions can help lead to more concept generation and more deep reasoning.

The few blocks that were observed in your team were met with deflection, rather than overcoming. Deflection implies accepting a block and suggesting alternative ideas. Overcoming implies rejecting a block and persisting in the original idea. Block-overcoming interactions can seem argumentative and are associated with deep reasoning through argumentation.

Discussion Questions:

How do you encourage your teammates' ideas?
Have you experienced argumentation as a means to deep reasoning in your team interactions?

B. Team Roles based on interaction responses observed

Supporter: Person A

Supporting behavior is key to maintaining group cohesion. In the Interaction Dynamics Notation, supporting behavior is identified through back-channeling behavior, positive acknowledgment, and indications of approval.

Questioner: Person B

Question-asking has been correlated with design performance in prior research studies. We identified the people who exhibited greater question-asking behavior as questioners.

Discussion Question: Do you sense these roles in your team? Who do you believe plays these roles in your team?

C. Balance of Contribution

Persons B and C were more active during the team interaction than person B.

We calculated the aggregate number of IDN responses, excluding humor and support, to arrive at the individual contribution of each team member.

	A	B	C
Contribution	23	47	48

Prior research on collective intelligence has indicated that balanced team contributions in general are correlated with higher performance. However, there is no research specific to design thinking teams on this topic.

Discussion Question: What do you think about the balance of individual contributions in your team? Are you okay with the current contribution levels?

Fig. 2 IDN instrument based feedback report

"... the experiment... gave us a very clear picture of how we work and interact with each other... Among the things that we are considering is transitioning from the "blocking method" to the "deep questioning method" of approaching the problem and see which one works best for us."

"I also was slightly skeptical about the CDR report given that it focused on the quantity of specific contributions from team members, disregarding quality completely."

"I also appreciated the analysis... about our group. The conclusion of its analysis was that I am more a questioner than someone who moves things. So I'm glad to know that and in the future I'll try sometimes to move things on instead of asking deep questions."

Overall, the research team noticed that most students accepted the feedback as a reflection of themselves as individuals, rather than one data point in the dynamic process of teamwork. The nature of the feedback report with an emphasis on individual contribution and numerical analysis versus sequence analysis could have contributed to this effect.

6.2 Study 2: In-Situ Human Mediated Delayed Feedback

Two student teams from ME310, a graduate level course in Engineering Design at Stanford University, were selected for this study. Video of the team's activity was recorded in-situ in the ME310 studio. The teams called the researchers to record their on-going activity when they were engaged in their design project discussions. Researchers analyzed the video using the IDN instrument and conveyed findings from the analysis back to the teams with a 2-week delay. Figure 3 shows screenshots from the video recording of the two delayed feedback teams.

This feedback session was followed by an interview in which the researchers asked about the teams' perception about the feedback. The teams reported the following perceptions about the feedback they received.

1. The teams felt the feedback was as expected, without any surprises. The students had noticed similar tendencies in their team behavior and the feedback validated what they had noticed before.
2. The teams felt the feedback was not useful since the conversation occurred 2 weeks prior to feedback. Most students did not remember the conversation.
3. When asked as to what kind of feedback the teams would consider meaningful, one of the teams mentioned they needed more information about specific methods for concept generation rather than feedback on their interactions.

When giving feedback, the research team noticed the following:

1. It was difficult to give normative feedback since the naturalistic conversations of the team did not pertain to a specific design phase such as concept generation or

Fig. 3 Video stills from the design team sessions recorded in-situ

decision-making. The conversations touched upon a number of aspects including coordinating logistics, generating concepts, and clarifying plans.

2. The researchers were cautious not to bias the team with normative feedback about beneficial or detrimental interaction patterns since the patterns studied earlier might not be applicable to the teams' context.

6.3 Study 3: Direct Real-Time Feedback

This study was conducted with two student teams in ME310, a graduate level course in Engineering Design. Real-time feedback was implemented by having the researchers sit next to the teams being analyzed. The researchers coded the interactions of the teams as they were occurring in an excel sheet on the Google Documents platform that was linked to a graph. This graph changed in real-time as the researchers coded IDN; the team could see the graph on a laptop screen set in front of them. These real-time sessions were video recorded as well, for future analysis. Figure 4 shows the real-time setup and Fig. 5 shows the graphical feedback display visible to the teams.

The real-time display showed a bar graph with three categories: generative behaviors, deep-reasoning behaviors, and team cohesion. We obtained these categories by combining several IDN symbols, as we believed that showing the IDN representation would be complicated for design teams to understand in real-time.

An interview with team, in which the researchers asked about the teams' perception of the feedback, followed the real-time feedback session. The teams reported the following:

1. The team members were aware of the display and they glanced at the display at times, but did not know what to do with the information displayed.
2. On a scale of 1–5, 1 being no difference, 5 being significant difference; the teams rated the display as 1 to 2. It made little to no difference to the on-going conversation.
3. A few team members mentioned they would prefer a personalized display that showed their behavior parameters rather than a team level display of interaction information.

Fig. 4 In-situ real-time feedback display for design teams

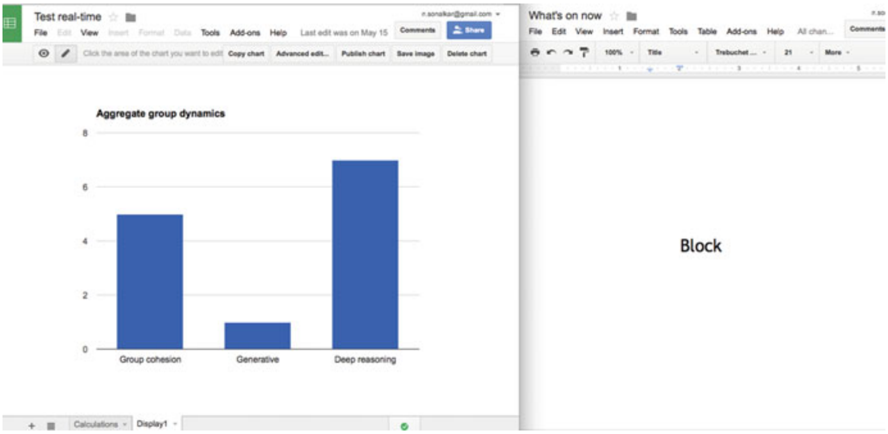


Fig. 5 The display showed a bar graph of three categories and in-the moment coding of IDN in the form of words flashed on the screen

4. One team reported that they had an implicit desire to see all the bar graphs grow equally. They felt unhappy if one of the bars was low.
5. One participant mentioned he would like to see a graphical display that showed both positive and negative progress. He felt if the feedback showed negative progress then it would be more relevant to the team.
6. Most team members felt they needed feedback that was actionable. The current feedback did not come with what they should do when they saw the parameters vary.

The research team when giving feedback noticed the following:

1. The researchers could just barely keep up with the conversation. It was important to be able to hear well, which was difficult since the studio environment was noisy. Hence it is likely that some complex symbols like 'block-overcoming' and 'yes and' were missed.

2. Due to heavy cognitive effort, the researchers could perform real-time IDN analysis for at the most up to 20 min at a time.

This study indicated that the design teams did not perceive real-time feedback to be useful, largely because they did not consider the feedback to be relevant and actionable. The real-time feedback was cumulative and the teams reported that perhaps negative feedback could be more relevant to them in real-time. Though teams did not perceive the feedback to be particularly useful, the study uncovered potential directions for further inquiry such as personalized feedback and variation in the graphical representations of the feedback.

6.4 IDN Instrument User-Interface Design Requirements

The three studies described in this section point to the following design requirements for developing a user interface for an IDN instrument.

1. Timeliness: The feedback needs to be given in a timely manner so that the team has the opportunity to understand it and act on it. Teams consider delayed feedback to be less useful. The studies indicate that feedback given just after a design session, or up to a 5-day period after the session, may be useful to the design teams.
2. Non-disruptive nature: The feedback should not disrupt the flow of the design team interactions unless the designers themselves or the severity of the situation call for it.
3. Lucidity: The feedback needs to be presented in a manner that designers can understand in a short period of time. The cognitive effort to understand feedback should not disrupt the ongoing design activity.

Besides the requirements for the IDN instrument user interface, the studies also revealed that the design teams receiving feedback need to be prepared beforehand. The design teams need to be given a mental model of how to do design with interaction feedback. The design teams need to have the skills to both understand the feedback and to act on it. Feedback cannot be presented as an afterthought; it should be integrated in the routine processes of design thinking.

7 Next Steps in Design Thinking Instrumentation

In this chapter, we described the on-going development of the IDN instrument for measuring design thinking team interactions. The first step in developing the instrument was to create a sensing system to convert actual team interactions into a representation that can be then compared with a measurement standard. The sensing system was successfully developed and was further used with design teams to generate a set of requirements for developing a user interface. However, while conducting these studies, it became clear that the next most significant step is the development of a measurement standard so that the instrument output given to users can perceived as relevant and useful.

Design thinking is a context dependent activity. Context parameters such as the nature of the problem, its domain, the expertise of team members, and the diversity of the team can potentially play an important role in influencing the team interaction behaviors that count towards innovative product outcomes in a design team. The measurement standard to be developed would need to take these context parameters into account. Therefore, the next step in developing a measurement standard is the characterization of design context and a descriptive model of high performance design team interactions associated with relevant context parameters.

The sensing system, a user interface, and a measurement standard that takes into account design context would together constitute a scientific instrument for design thinking. Instrumenting design thinking teams has the potential to transform design thinking from a purely heuristics driven activity to an activity where scientific knowledge is employed to provide relevant feedback that augments human ability to achieve innovative outcomes reliably and efficiently.

References

Baird D (2004) Thing knowledge: a philosophy of scientific instruments. University of California Press, Berkeley, CA

Brandt PA (2004) Spaces, domains, and meaning. Essays in cognitive semiotics, European semiotics series, vol. 4. Peter Lang AG, Bern

Brown T (2008) Design thinking. Harv Bus Rev 86(6):84

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46

Dorst K (2011) The core of 'design thinking' and its application. Des Stud 32(6):521–532

Hubert L (1977) Kappa revisited. Psychol Bull 84(2):289

Johansson-Sköldberg U, Woodilla J, Çetinkaya M (2013) Design thinking: past, present and possible futures. Creativity Innov Manag 22(2):121–146

Kolko J (2015) Design thinking comes of age. Harvard Business Review 93(9):66–71

Lakoff G, Johnson M (2008) Metaphors we live by. University of Chicago Press, Chicago, IL

Rauth I, Köppen E, Jobst B, Meinel C (2010) Design thinking: an educational model towards creative confidence. In: DS 66–2: Proceedings of the 1st International Conference on Design Creativity, Kobe, Japan

Sonalkar N, Mabogunje A, Leifer L (2013) Developing a visual representation to characterize moment-to-moment concept generation in design teams. Int J Des Creativity Innov 1 (2):93–108

Talmy L (1988) Force dynamics in language and cognition. Cogn Sci 12(1):49–100

Yujian L, Bo L (2007) A normalized Levenshtein distance metric. IEEE Trans Pattern Anal Mach Intell 29(6):1091–1095

Ziółko B, Gałka J, Jadczyk T, Skurzok D (2010) Modified weighted Levenshtein distance in automatic speech recognition. In: Proceeding of 16th Conference on Applications of Mathematics, Krynica, Poland

The Topic Markup Scheme and the Knowledge Handling Notation: Complementary Instruments to Measure Knowledge Creation in Design Conversations

Axel Menning, Andrea Scheer, Claudia Nicolai, and Ulrich Weinberg

Abstract The focus of this book chapter is the introduction of two complementary instruments, the Topic Markup Scheme (TMS) and the Knowledge Handling Notation (KHN) to analyze design conversations. Both instruments will be applied to a design conversation sample. TMS offers a diagnostic procedure that is capable of describing the topical structure of a conversation and its move-to-move coherence. KHN describes on the move-to-move level how innovation teams generate and share knowledge. The output of both instruments, in the form of strings of symbols can be used for sequence analysis and pattern detection of team dynamics. Together, the outcomes nurture the understanding of knowledge creation in and through design conversations in innovation teams.

1 Introduction

Designers use external representations, such as sketches, models, or even reenactments to create ideas and knowledge around a certain design issue (Schön 1983; Goldschmidt 1991; Suwa and Tversky 1997). For example, through sketching a thing, detailed questions about that thing arise. Producing external representations triggers interaction with one's own knowledge. As Schön puts it, it is like having a conversation with material and situations. This form of knowledge creation is well-represented in the Design Thinking principles and methods taught and practiced at Stanford University and Hasso Plattner Institute, for example through the emphasis on visualization and prototyping (Plattner et al. 2009). Another well-known element in design thinking is collaboration in interdisciplinary teams. But how do design teams create knowledge through collaboration and—more

The original version of this chapter was revised. An erratum to this chapter can be found at DOI 10.1007/978-3-319-40382-3_18.

A. Menning (✉) • A. Scheer • C. Nicolai • U. Weinberg
HPI School of Design Thinking, Hasso Plattner Institute for Software Systems Engineering, Campus Griebnitzsee, 14482 Potsdam, Germany
e-mail: axel.menning@hpi.de

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_16

specifically—through design conversations? The two instruments introduced in this chapter will contribute to the design theory of knowledge creation through design conversations.

The progress of conversations and the topics exchanged in conversations are highly context dependent (van Dijk 2008; Denzin and Keller 1981). In design conversations such a context dependency exists for example with the notion of "wicked problems" (Rittel in Protzen and Harris 2010). In consequence, the goal determination in design teams is characterized by highly iterative adjustments and reframing. Goal setting "is often carried out throughout the design process and not only at the beginning". This leads to a constantly changing topical focus in conversations. For example in his article "The Reasoning of Designers" Horst Rittel states that the designer's focus "alternates continually from small component parts, back to the whole problem, and back to other details" (Rittel 1987). Many more unique context parameters exist for design conversations. Some of these parameters are covered by the design thinking conversation criteria (Scheer et al. 2012). Accordingly, design thinking conversations are collaborative per se because design thinking teams consist often of up to six team members with different disciplinary backgrounds. Design thinking teams fall into the paradigm of constructivism as knowledge is collaboratively constructed around topics. This approach requires self-regulation of participants to manage knowledge handling and topical progress. Due to these unique context parameters we argue, that design conversations carry unique features in terms of knowledge creation. The two complementary instruments—the Knowledge Handling Notation (KHN, Scheer et al. 2014) and the Topic Markup Scheme (TMS, Menning et al. 2015) will contribute to make these features visible.

The conversation analysis with both instruments works on the highest possible resolution: the move-to-move level. A move is a person's verbal contribution to a conversation at a certain time. A move begins when the speaker begins to speak and ends when the speaker ends her or his articulation deliberately or is interrupted. KHN describes how design teams handle knowledge and TMS determines the topical relatedness between a move at a certain time (m_n) and another. More precisely, TMS provides sequential descriptions of:

(A) move-to-move coherence

- perceived topical relatedness of m_n to m_{n-1}
- perceived topical relatedness of m_n to m_k (with m_k being a move prior to m_{n-1})

(B) global coherence

- perceived topical relatedness of one design issue to its preceding design issue (with an issue being a set of moves).

1.1 Measuring Level I: Knowledge Handling of Design Thinking-Teams

Knowledge handling is an activity in which knowledge is shared and created in an attempt to find innovative solutions to a given challenge. This is usually apparent when people come together to work as a team. Especially when people have different disciplinary backgrounds, knowledge handling becomes key to reaching a common ground of understanding (Bromme et al. 2004) and is the basis for teamwork (Katzenbach 1993). When working as a team, people converse with each other, and in discussing their beliefs jointly share and create knowledge towards a shared goal.

Team conversations are characterized by a team interacting to solve a challenge, understand a problem, find an answer to a question, and create a solution, among other activities. Team conversations on a verbal level primarily function as interactions to share and create knowledge about a specific topic, which is beneficial for solving a challenge. Doing so, team members switch between created topics that also feed each other mutually.

However, there is not much known about what exactly happens in such team conversations or how participants perform knowledge sharing and creation. In perspectives of conversation analysis and team interaction dynamics, the knowledge handling process still seems to be a black box. We aim to illuminate this black box in order to find out more about the underlying key activities, patterns, and dynamics of team conversations. A first step towards an understanding of knowledge handling in conversation is to capture knowledge sharing and creating activities, as well as the dynamics that define the variety of conversations with different participants.

1.2 Measuring Level II: Coherence Measures in Design Conversations

The description of move-to-move coherence and global coherence is a subject of discourse structure theory. Grosz's and Sidner's (1986) work on discourse structure differentiates between the overall communicative goal, which they call Discourse Purpose (DP), and its constituents, the Discourse Segment Purposes (DSPs). In order to achieve DP, discourse segments should be meaningfully aligned (global coherence). In order to achieve a DSP, utterances in a move should be meaningfully aligned (local coherence) and two subsequent moves should be meaningfully aligned, which we call "move-to-move coherence".

We argue that due to the nature of designing and the circumstances in which designing takes place, design conversations often lack an explicit focus or communicative goal. This effects the coherence of design conversations. If coherence gaps exist, the discourse participants are forced to infer meaning to achieve coherence. This associative act of inferring tends to trigger ideas. Thus coherence gaps can be seen as potentially positive stimuli for ideas in design conversations. According to Suwa and Tversky, focus shifts in design conversations "allow for a lateral variety of design topics/ideas. . .” (Suwa and Tversky 1997). This argument can be backed up with creative behaviour experiments on the phenomenon of fixation. Fixation refers to the mental state of individuals who are unable to move beyond an idea or set of ideas to produce new ideas (Howard-Jones and Murray 2003). It was suggested by Finke et al. (1992) that broadening the focus of attention supports overcoming fixation which would lead to a greater variety of ideas. Thus, coherence measures in design conversations are a promising field to look into if one wants to find out more about the evolution of ideas.

TMS will provide insights into the question of at what stage and to what extent topical focus and diffusion are beneficial (or counter-productive).

2 Complementary Diagnostic Instruments: KHN and TMS

In the context of team interaction Sonalkar (2012) refers to process-based measures as diagnostic procedures. A diagnostic procedure generally reduces the complexity of the object under investigation to a set of primitives. In our research we aim to reduce the complexity of natural language and interpersonal communication to a minimal set of symbols that represents either knowledge handling or the topical structure of design conversations. The objective gains both a higher level of abstraction and a more precise view of these specific modes (pattern recognition). Both instruments were developed and validated through iterative analysis rounds of video data based on *grounded theory* (Glaser and Strauss 1967). Whenever it was not possible to assign a speakers’s turn to an existing mode of interaction, a new mode was created. This new set of interactions was then reduced to more general categories. As Martin and Turner (1986) state: “A key element of *grounded theory* is identifying a slightly higher level of abstraction—higher than the data itself.” After several rounds of applied creation and validation of categories, the selected categories were described with test cases in a coding manual. This coding manual was then used to move coding decisions towards a formal set of rules. A tight relation between symbols and text base leads to a high inter-coder agreement. We also found out that the transcription of the coding manual to a decision tree format improved inter-coder agreement.

2.1 The Knowledge Handling Notation (KHN)

2.1.1 Description

The Knowledge Handling Notation is a visual representation of knowledge handling activities in interpersonal interaction. It consists of five symbols (including N/A). Table 1 lists the different symbols and the category of knowledge handling they represent.

Table 1 Overview of KHN coding scheme

Category	Symbol	Name	Description	Example
Push	o	**Externalization of knowledge**	**Articulated justified personal belief, knowledge transformation tacit to explicit through personal statements**	**"It's cold outside." "It's −5 °C."**
	oo	*Combination of knowledge*	*Synthesis of articulated justified personal beliefs, knowledge transformation explicit to explicit through sorting, adding, recategorizing, recontextualizing of explicit personal statements*	*"Oh yes, −5 °C is very cold."*
Pull	>o	**Trigger for externalization**	**Calling for externalization, e.g. through questions, imperatives**	**"How cold is it?"**
	>oo	*Trigger for combination*	*Calling for combination, e.g. through questions, imperatives*	*"Tell me if that is cold for a German winter!"*
	/	Not applicable, N/A	No transformation of knowledge involved	"Yeah. . ."

Bold Sharing mode
Italic Creating mode

In KHN, verbal speaker articulations are interpreted and assigned symbols to create a descriptive visual model of knowledge handling. The assignment of symbols is conducted based on the speaker, which means on the basis of what the expression is intended to be by the person making it. The Knowledge Handling Notation is a mixed-method instrument for capturing knowledge handling activities in conversation. It aims to enable a quantitative analysis with non-standardized data. Non-standardized video data of team conversations, including transcripts, is converted into discrete data representing knowledge handling activities. This is done using the Knowledge Handling Notation. KHN assigns codes to speaker turns in non-standardized data. The codes are developed based on Nonaka's notions of externalization and combination in his SECI model (Nonaka 1994). Once the data is coded with KHN, the researcher can statistically analyze team conversation in terms of knowledge handling activities. Inclusion criteria for applying KHN code to video data or transcript are that:

1. The speaker's turn is verbally expressed in the conversation.
2. The speaker's turn can be understood by the coder acoustically and language-wise.
3. The speaker's turn concerns a transformation of knowledge of the team or individuals in the team. This might be:

   (a) a transformation from tacit to explicit knowledge
   (b) a transformation from explicit to explicit knowledge
   (c) a call for transformation.

2.1.2 Limitations of KHN

KHN aims to represent knowledge sharing and creation through verbal interaction in design thinking team conversations. It attempts to model the mutual constitution of externalization and combination of explicit knowledge.

Through pattern analysis of coded conversation sequences, one might find different forms of transformative processes that are described by Stahl as articulation, discussion, clarification, negotiation, and formalization (Stahl 2000). However, KHN does not represent the outcomes of these processes. In a design thinking setting, such outcomes might be equal to the outcomes of the design thinking phases, e.g. shared insights, point of view, prototypes or other artifacts.

The KHN output is a string of symbols representing knowledge handling of a given conversation. It represents what and how many activities were involved in sharing and creating knowledge, as well as the speaker turns and time stamps. It does not represent if the knowledge handling activities were related to one topic of the conversation or switched between many topics. This could be crucial when analyzing knowledge sharing and creating outcomes of conversations. The Topic Markup Scheme as an additional layer for conversation analysis is useful here.

2.1.3 Future Work

Until now KHN was applied and tested with a limited dataset. Certain cases are still hard to code. Therefore, we are creating a case compendium with a collection of codings for many different kinds of cases. Further, we are developing a coding test that helps the coder to check the coding decision for accuracy.

2.2 Topic Markup Scheme

The TMS procedure consists of a coding scheme, rules, and a visualization scheme. TMS is related to Goldschmidt's research on the construction and representation of topics and their alignment in design conversations. Goldschmidt developed Linkography, which is a method to represent topical links in conversations (Goldschmidt 1990, 2014). While a Linkograph captures moves and their environment of referencing moves binary (linked or not linked), TMS describes the link between two consecutive moves (called transition state) with four different characteristics: *continuation*, *drift*, *integration*, *jump*. In the following paragraph the TMS procedure will be introduced by listing the rules to assign the center of a move as well as the rules to determine the transition states between two subsequent centers.

2.2.1 The Center of a Move

TMS builds on a centering approach that was introduced by Grosz, Joshi, and Weinstein as *centering theory* (CT) (Grosz et al. 1983, 1994). CT is based on the assumption that each utterance in a discourse segment has one entity, which is topically more central than others. This entity is called a center. While CT operates on the level of utterances, a rather unspecified discourse entity (Poesio et al. 2000), TMS, operates on the level of moves. The length of a move is well-defined but can vary from a single word to multi-sentence statements. In order to meet this variety with a flexible assessment, move-centers $C(m_n)$ are defined as follows.

1. Each move has exactly one center C
2. A center can either be explicit, $C_e(m_n)$, or implicit $C_i(m_n)$

(a) A center is explicit when there is a direct topical overlap from m_n to m_{n-1}. This topical overlap can be either a direct syntactic overlap, meaning that for example a noun in m_n is matched with a content word in the previous move m_{n-1} (noun overlap, stem overlap), or a semantic overlap. Take the following example:
Speaker A: *So we're going to share our perspective, right? So which...*
Speaker B: *I think we should* ***explain*** *to them the clown thing...*
In this case we consider > explain < as C_e because it is semantically similar to > share < .
(b) If m_n has no C_e, we define that m_n has a C_i. The $C_i(m_n)$ is either a content word or phrase that shows syntactic or semantic overlap to a C prior to m_{n-1} or we define the $C_i(m)$ similar to the preferred center of Centering Theory. In this case, it is the highest ranked entity according to salience ranking.
(c) If m_n has no content word, which is the case for one word moves (e.g., "yeah"), we define that the last $C_{e,i}(m_k)$ gets assigned.

2.2.2 TMS Transition States

Based on the annotation of move centers, the TMS transition states can be determined. The TMS transition states show and characterize the transition of two move centers and thus indicate how design teams treat design issues. A string of transition states is the raw outcome of TMS. The possible transition states are called *continuation*, *drift*, *integration* and *jump*. Figure 1 presents an overview of the determination of transition states.

The determination of TMS transition states is thus based on the following rules. All four rules are accompanied by examples. In the example transcript, the center is highlighted with a black background and its referent is underlined and bold.

(1) Continuation $(m_{n-1}, m_n) := C_e(m_n) = C_{e,i}(m_{n-1})$
A move (m_n) continues a preceding move (m_{n-1}) if the center is explicit and shares the same semantic value with $C_{e,i}(m_{n-1})$ (Fig. 2).

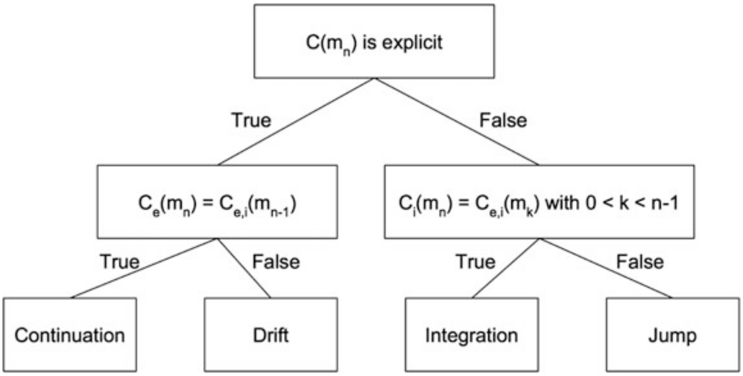


Fig. 1 Overview of the determination of transition states

ID	Move with Center
13	And **routines**
14	Routines That make them this kind of feeling of safeness.

Continuation
[(C_e14 = Routines) = ($C_{e,i}$13 = routines)]

Fig. 2 Example transcript for the transition state *continuation*

14	Routines That make them this kind of feeling of **safeness.**
15	Gives them safeness.

Drift
[(C_e15 = safeness) ≠ ($C_{e,i}$14 = Routines)]

Fig. 3 Example transcript for the transition state *drift*

121	Just, just... So this is the **work... ethics**. And then what this will do to the homeless person.
131	That's up here, work ethics. It's still... yeah.

Integration
[(C_i131 = work ethics) = ($C_{e,i}$121 = work ethics)]

Fig. 4 Example transcript for the transition state *integration*

(2) Drift $(m_{n-1}, m_n) := C_e(m_n) \neq C_{e,i}(m_{n-1})$
We define m_n as drifting if its center is explicit but does not share the same semantic value with $C_{e,i}(m_{n-1})$ but with any other content word of m_{n-1} (Fig. 3).

(3) Integration $(m_{n-1}, m_n) := C_i(m_n) \neq C_{e,i}(m_{n-1}) \wedge C_i(m_n) = C_{e,i}(m_k)$ with $0 < k < n-1$
m_n integrates if it is not meeting the conditions for *continuation* or *drift* but if the $C_i(m_n)$ relates to a center of a move prior to m_{n-1} (Fig. 4).

(4) Jump $(m_{n-1}, m_n) := C_i(m_n) \neq C_{e,i}(m_{n-1}) \wedge C_i(m_n) \neq C_{e,i}(m_k)$ with $0 < k < n-1$
If conditions for *continuation*, *drift* and *integration* are not met, m_{n-1} is discontinued and a new topic is introduces with m_n (Fig. 5).

31	That... You've got a structure of the day and it kind of stabilizes your...
32	We already have something like a customer relationship here, so...

Jump
$[(C_i32 = \text{customer relation}) \neq (C_{e,i}31 = \text{structure})]$

Fig. 5 Example transcript for the transition state *jump*

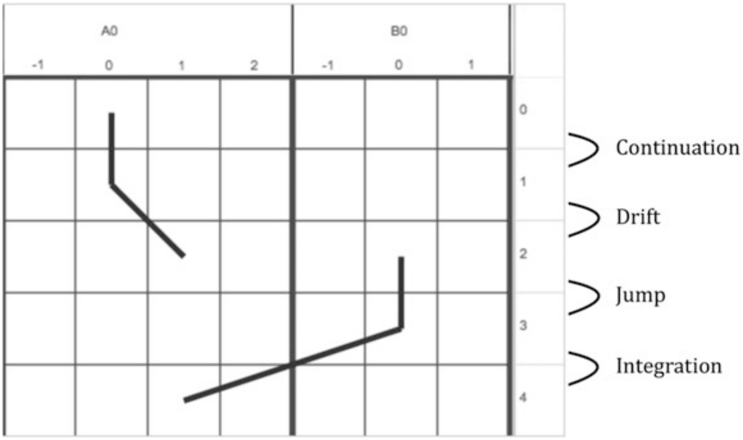


Fig. 6 Environment for visual coding and representation of transition states

After the transition states are determined, the string of moves, move parameters, and transition states can be further analyzed using statistical methods. The transition states *jump* and *drift* can be subsumed to have divergent tendencies because in both cases a new or changed topic gets introduced. The transition states *continuation* and *integration* have convergent tendencies because a certain topic gets treated and/or integrated with another one. In addition, the processed transcript can be represented in the form of topic threads.

2.2.3 Visual Representation

The visual coding is performed in a two dimensional lattice as shown in Fig. 6. The four transition states (*continuation*, *drift*, *jump*, *integration*) are illustrated by drawing a line between two moves (two rows). This line represents a topic thread.

A column (e.g. column A in Fig. 6) represents one issue with a certain topic width. The topic width depends on the number of moves that are coded as *drifts*. We count the topic columns alphabetically and the topic sub columns numerically. The row number on the right side counts down linearly and represents the move number. The continuation of a topic connects the occurring move with the latest move and is represented by a straight line downwards (e.g. A00→A01). The topic drift connects a move with the last move and ends one cell downwards and creates a new sub-topic column on the right side of the current topic column (e.g. A01→A12). A topic jump means the discontinuation of the current topic thread and opens a new topic one column to the right. It is represented as a straight line downwards in the new topic column (e.g. B02→B03). An *integration* move connects to the latest move and ends one cell downwards in the column where the move of the integrated topic has ended before (e.g. B03→A14). We assume that the representation of topic threads enables intuitive comprehension of the coding outcome. This would be beneficial for design teams and design educators to instantly assess the coherence of a conversation.

2.2.4 Limitations and Future Work of TMS

A limitation of the current method is the infinite *topic width*. At the current state of the method, it could be misleading that long topic drifts produce one thread in one column that actually consists of several topic values. We want to contain this issue by defining threshold rules to transfer a sequence of topic drifts to a topic jump. TMS (and KHN) focus mainly on verbal aspects of team interaction while meaning making in conversations is highly influenced by nonverbal communication and spatial relationships (e.g. gestures and positions of team members). We also know that design conversation contain many deictics like ‘here’, ‘over there’, ‘this’, and ‘that’ (Glock 2001; Schön 1983.) Such deictic expressions cannot be fully understood without their referents. We want to overcome this by working closely with the video recordings. Moreover the analysis consists partly of a subjective assessment of the semantic relatedness of two moves. In our future work, we want to further support the human analysts with automatic suggestions regarding the semantic relatedness.

2.2.5 Coding Samples of KHN and TMS

In this section we will demonstrate the coding procedure of KHN and TMS applied to an example data set. The data consists of in situ video recordings and transcripts from two different design tasks carried out by a student team at the HPI School of Design Thinking (Potsdam). The design thinking team worked on a project for 6 days with the challenge of “improving the life of homeless people.” The students come from different academic backgrounds with different experience in designing. Figure 7 is a video still of the general team work set up.

For the example dataset we chose two contrary episodes in terms of divergent and convergent activities. We chose an episode of day 2, where the overall goal was the unpacking of the experiences during field research (interviews and observations). And we chose an episode of day 3, where the overall goal was the synthesis of collected data. We selected the two episodes by asking the team after their design sessions to indicate their most meaningful episode of the day. Both key episodes got

Fig. 7 Screen shot (split screen) of a recorded design conversation

translated, and parsed for the move attributes speaker, timestamp. The KHN coding sample results from activities of day 2, the TMS coding samples represent data from day 2 and 3.

2.3 KHN Coding Sample

The 10 turns in the coding sample above (Table 2) represent an episode of the synthesis phase. The team tried to focus on the aspect of *work ethics* and how they impact *homeless people*. Turn 1 started with a trigger for combination through calling for insights that might result when matching *work ethics* and *homelessness*. Speaker C starts right off with an externalization of his justified belief that *structure* would be related to *work ethics* in turn 2. Turns 3 to 8 combine in the area of *structure* by adding aspects that further define what *structure* might mean for *homeless people* (e.g., *daily routines* and *security*). Speaker D externalizes a new justified belief of *purpose* as also related to *work ethics* in turn 9. Though speaker E confirms this belief, he does not transform that knowledge further and it is coded as *not applicable*.

The sample started off with a call to get insights through relating work ethics with homelessness. During the course of 10 turns, the team came to the conclusion that work ethics create structure, security and purpose for homeless people (Fig. 8).

In the figure above, only the output of KHN codes is shown. It is visible that a trigger and externalization preceded a row of combinations.

Table 2 KHN sample coding

Turn	Speaker	Transcript	KHN	Comments
1	B	Just, just. . . So this is the work. . . ethics. And then what this will do to the homeless person.	>oo	Pull: call for synthesis about the relation between work ethics and homeless people
2	C	Ok, umm, it gives them, umm, structure. Umm. . . structure like, umm, err. . .	o	Push: justified belief about work ethics giving structure
3	D	It structures the day.	oo	Push: synthesis of turn 2, structure means structuring the day
4	C	It structures the day. That it looks more like it's daily. . . life.	oo	Push: synthesis of turn 3, a structured day means daily life
5	D	And routines.	oo	Push: synthesis of 4, structuring daily life means routines
6	C	Routines. That makes for them this kind of feeling of safeness.	oo	Push: synthesis of turn 6, routines mean safety feeling
7	B	Gives them safeness.	oo	Push: synthesis of turn 6, routines mean safety feeling
8	C	Security. . .	oo	Push: synthesis of turn 6, routines mean security
9	D	And also a purpose.	o	Push: justified belief about work ethics giving a purpose
10	E	M-hm. (confirming)	/	No transformation of knowledge involved

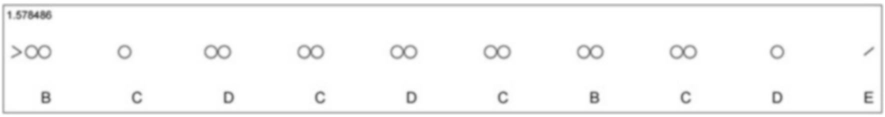


Fig. 8 KHN output of coding sample

2.4 *Outcome Measurement*

In order to find out how well a team performs in respect to their design challenge, we used a concept score as a measurement to evaluate the team's final outcome. Based on the creativity score concept introduced by Shah and Vargas-Hernandez (2002), we adapted the concept score questionnaire to fit the analyzed team's challenges. The concept score consists of three parameters to measure the outcome of a team: utility (to what extent does the outcome meet the following desired specifications: (a) helpfulness, (b) desirability for the user and (c) feasibility and (d) implement ability for the project partner), novelty (a) novelty of outcome, (b) novelty of created experience and (c) originality of overall concept for stated problem) and need relevance (does the formulated need reflect an important pain point; is the need surprising and does it address an important aspect of the challenge?).

The outcomes of the filmed teams were evaluated by Design Thinking coaches, another Design Thinking partner team working independently on the same challenge, and by the project partners who had proposed the challenge to the teams. In the fourth research quarter, independent coaches who hadn't coached the teams carried out the evaluation of the filmed teams' outcome for the 6 day project. To guarantee a direct comparison of the outcome of the two independently working teams, the coaches were asked to evaluate both teams immediately after watching both videos of their final presentations and reading through both documentations.

2.5 Analysis with TMS

Both video recording transcripts from day 2 (Task: Unpacking of field research results) and day 3 (Task: Synthesis field research data) were independently analyzed by two researchers. According to the TMS procedure as presented in 2.2, each researcher began with assigning a center to each move and marking up reference words or phrases in the preceding text (if existent). Afterwards, the TMS transition states: *continuation*, *drift*, *integration*, *jump*, were determined. Both results were compared. In the case of deviation, both researchers were asked to find a consensus by revisiting the transcript. Table 3 shows the absolute occurrences for each transition state type in the indicated phase. Because the key episodes have a different absolute number of moves, the percentage number of each move type was calculated.

Figure 9 shows the total occurrence of transition states in both episodes.

Interestingly, the grouped bar chart shows for the key episode of unpacking a majority of the symbols *drift* and *jump*. The design task unpacking can be seen as a divergent activity and also the transition states *drift* and *integration* have divergent characteristics as they indicate an enrichment of topics. The synthesis task on the other hand is a convergent task because the goal of the students is to turn a large amount of data from the field research into a single sentence called "point of view". In this task the transition states continuation and integration are dominant. *Continuation* and *integration* have convergent tendencies as they tend to solidify a topic instead of broadening it.

Table 3 Absolute and percentage number of moves

Move type	Unpacking phase: Absolute number	Unpacking phase: Percentage number	Synthesis phase: Absolute number	Synthesis phase: Percentage number
Continuation	25	53.2	35	61.4
Drift	8	17.0	6	10.5
Integration	7	14.9	11	19.3
Jump	7	14.9	5	8.8
Total	100	100	107	100

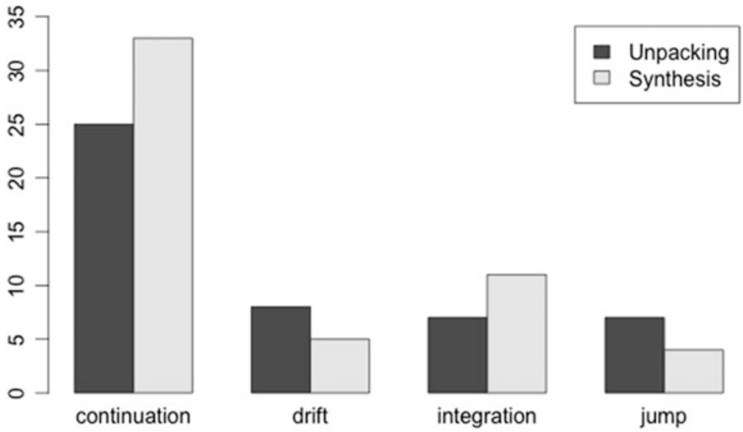


Fig. 9 Grouped bar chart to compare the total occurrence of transition states

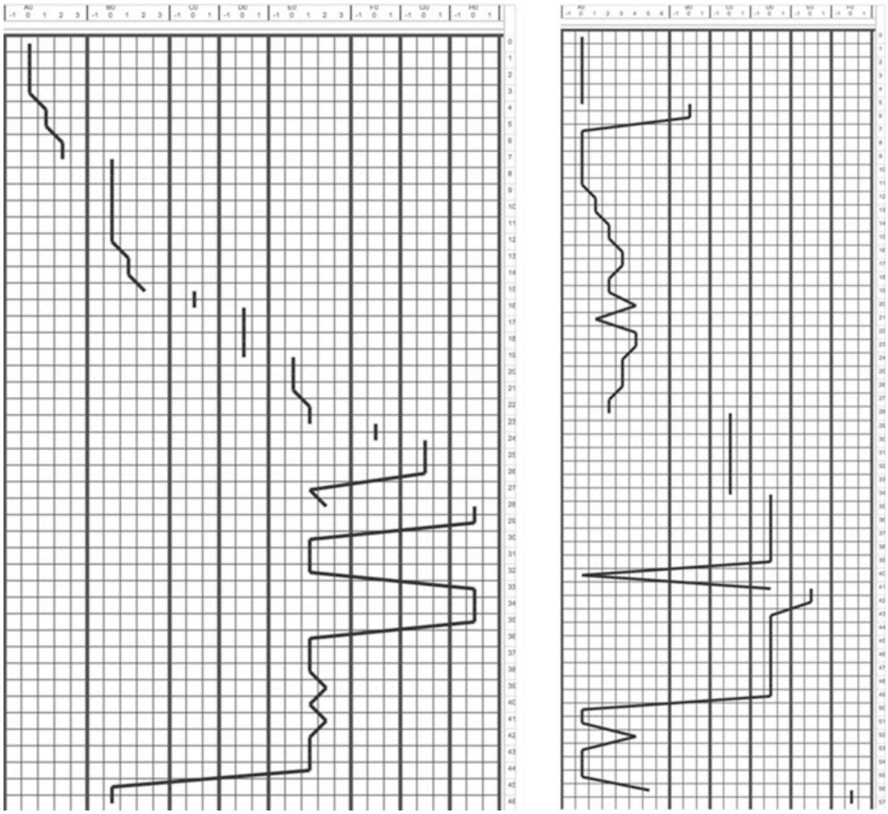


Fig. 10 TMS Topic thread of unpacking episode (*left*) and synthesis episode (*right*)

In addition, the two resultant TMS strings (one for each design task) were visualized as topic threads, using the TMS representation method (cf. Sect. 2.2.4). The topic thread begins on the upper left side and moves downwards. The horizontal lines mark the borders of different discourse segments or design issues. A new design issue gets introduced by the transition state *jump* (Fig. 10).

3 Conclusion

Both, KHN and TMS have been shown to represent knowledge handling and topical progress in conversations. The example output of the TMS (as presented in Sect. 3.2) was confirmed by matching the convergent of divergent tendencies of the transition states with the convergent of divergent tendencies of the two different design tasks. The output of KHN and TMS is sequential, in the form of strings of symbols. The symbols represent a certain state with certain characteristics. Next to global distribution and ratio measures, we assume that sequential analysis will be insightful. By identifying certain events in a conversation, researchers can trace back the conversation history, identify patterns, and reason the development of these events. Because conversation depends on many different context parameters, the output of a single instrument has its limits in representing what is happening in a conversation. Take the following example showing in each case the analysis of the same excerpt:

(7) KHN (o,o,o),
TMS (continuation, continuation, continuation)
(8) KHN (o,o,o),
TMS (continuation, jump, continuation)

The interpretation of the KHN outcome on a sequential level would different for example 7 and 8. This would be the case because the topic of 8 changes, while it continues in example 7. This means the outcome can be mapped to another conversation context in order to achieve a greater accuracy of the diagnosis. Therefore depending on the focus of analysis, the analyst would have to use other complementary lenses to mark sequences of interest within the output of the KHN or TMS output. For example, analyzing patterns that lead to idea development, would call for a method or instrument that enables the identification of ideas. Therefore KHN and TMS are developed as complementary instruments. Another focus of interest is the analysis of KHN or TMS in connection with the Interaction Dynamics Notation (Sonalkar 2012). The structured analysis of the output of KHN or TMS will nurture the understanding of knowledge creation in and through design conversations.

References

Bromme R, Jucks R, Rambow R (2004) Experten-Laien-Kommunikation im Wissensmanagement. In: Reinmann G, Mandl H (eds) Der Mensch im Wissensmanagement: Psychologische Konzepte zum besseren Verständnis und Umgang mit Wissen. Hogrefe, Göttingen, pp 176–188

Denzin NK, Keller CM (1981) Frame analysis reconsidered. Contemp Sociol 10(1):52–60
Finke, R. A., Ward, T. B., & Smith, S. M. (1992). *Creative cognition: Theory, research, and applications*. Cambridge, MA: MIT Press.
Glaser BG, Strauss A (1967) The discovery of grounded theory: strategies for qualitative research. Aldine, Chicago, IL
Glock F (2001) Design work in contexts—contexts in design work. In: Lloyd P, Christiaans H (eds) Proceedings designing in context. Delft University Press, Delft, The Netherlands, pp 199–127
Goldschmidt G (1990) Linkography: assessing design productivity. In: Trappl R (ed) Cyberbetics and system'90. World Scientific, Singapore, pp 291–298
Goldschmidt G (1991) The dialectics of sketching. Creativity Research Journal 4(2):123–143
Goldschmidt G (2014) Linkography—unfolding the design process, 1st edn. The MIT Press, Cambridge, MA
Grosz BJ, Sidner CL (1986) Attentions, intentions and the structure of discourse. Comput Linguist 12:175–204
Grosz BJ, Joshi AK, Weinstein S (1983) Providing a unified account of definite noun phrases in discourse. In: Proceedings of the 21st annual meeting on association for computational linguistics, association for computational linguistics, pp 44–50
Grosz BJ, Joshi AK, Weinstein S (1994) Centering: a framework for modeling the local coherence of discourse. Technical Report, University of Pennsylvania, Department of Computer and Information Science
Howard-Jones, P. A., & Murray, S. (2003). Ideational productivity, focus of attention, and context. *Creativity Research Journal, 15*(2/3), 153. doi:10.1080/10400419.2003.9651409.
Katzenbach JR, Smith DK (1993) The wisdom of teams: creating the high-performance organization, McGraw-Hill management series. Harvard Business Press, Boston, MA
Martin PY, Turner BA (1986) Grounded theory and organizational research. J Appl Behav Sci 22 (2):141–157
Menning A, Scheer A, Meier B, Nicolai C (2015) Designing as weaving topics: coding topic threads in conversations. Conference paper for IASDR 2015, Brisbane, Australia
Nonaka I (1994) A dynamic theory of organizational knowledge creation. Organ Sci 5(1):14–37
Plattner H, Meinel C, Weinberg U (2009) Design Thinking Innovation lernen Ideenwelten öffnen. mi-Wirtschaftsbuch, Finanzbuch Verlag GmbH, Munich
Poesio M, Cheng H, Henschel R, Hitzeman J, Kibble R, Stevenson R (2000) Specifying the parameters of centering theory: a corpus-based evaluation using text from application-oriented domains. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics
Protzen JP, Harris DJ (2010) The universe of design: Horst Rittels theories of design and planning. Routledge Chapman and Hall, New York
Rittel H (1987) The reasoning of designers. International congress on planning and design theory in Boston, August 1987
Scheer A, Noweski C, Meinel C (2012) Transforming constructivist learning into action: design thinking in education. Des Technol Educ 17(3):8–19
Scheer A, Menning A, von Helldorff E, Rhinow H, Nicolai C (2014) The knowledge handling notation: building an interface to enable design conversation analysis. Design Thinking Research Symposium 10, Purdue, Lafayette, IN
Schön DA (1983) The reflective practitioner: how professionals think in action. Basic Books, New York
Shah JJ, Vargas-Hernandez N (2002) Metrics for measuring ideation effectiveness. Des Stud 24 (2):111–134
Sonalkar N (2012) A visual representation to characterize moment-to-moment concept generation through interpersonal interactions in engineering design teams. PhD Thesis, Stanford University

Stahl G (2000) A model of collaborative knowledge-building. In: Fishman B, O'Connor-Divelbiss S (eds) Fourth international conference of the learning sciences. Erlbaum, Mahwah, NJ, pp 70–77

Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Design Studies 18(4):385–403

van Dijk TA (1977) Sentence topic and discourse topic. Pap Slav Philol 1:49–61

van Dijk TA (2008) Discourse and context: a sociocognitive approach. Cambridge University Press, Cambridge

Developing Novel Neuroimaging Paradigm to Assess Neural Correlates of Improvisation and Creative Thinking Using fMRI

Manish Saggar, Lindsay C. Chromik, Adam Royalty, Grace Hawthorne, and Allan L. Reiss

Abstract The ability to produce novel yet appropriate (or useful) outcomes is broadly defined as creativity. Given the cultural significance of creativity, researchers have been attempting to uncover brain mechanisms underlying creative thinking since the early 1960s. However, several methodological issues have restricted researchers in uncovering the brain basis for creativity and previous neuroimaging studies have largely produced varied findings, with little overlap. Some of the methodological issues that could account for the large variance in results include treating creativity as a unitary construct, assessing creativity in a test-like environment, as well as explicitly prompting participants to "switch-on" creativity during certain parts of the experiment. To partly mitigate some of these issues, we recently developed a novel game-like and creativity-conducive neuroimaging paradigm that was employed to assess neural correlates of spontaneous improvisation and figural creativity in healthy adults (Saggar Sci Rep 5:10894, 2015). In this chapter, we provide a brief overview of the current state of neuroscience research focused on creativity. We also provide insights regarding our experimental design, challenges faced during prototyping as well as a summary of our results. Lastly, building upon our novel paradigm, we provide pointers to future work for assessing neural correlates of creative capacity enhancement and team creativity.

M. Saggar (✉) • L.C. Chromik • A.L. Reiss
Center for Interdisciplinary Brain Sciences Research, Stanford University School of Medicine, 401 Quarry Road, Stanford, CA 94305-5795, USA
e-mail: saggar@stanford.edu

A. Royalty • G. Hawthorne
Hasso Plattner Institute of Design (d.school), Building 550, 416 Escondido Mall, Stanford, CA 94305-3086, USA

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation,
DOI 10.1007/978-3-319-40382-3_17

1 Introduction

Creativity can be defined as the process of creating unique but meaningful solutions (Ellamil et al. 2012). Creativity is considered a necessary component for success in a wide range of areas and is considered an underlying factor for innovation and forward progress (Dietrich and Kanso 2010). Creativity is also considered an essential part of mental health (Srivastava and Ketter 2010; Torrance 1968). However, despite the importance and usefulness of creativity, it has been understudied compared to other psychological phenomena (Abraham 2013; Dietrich and Kanso 2010). Given the extent to which creativity has been implicated in success, productivity, and mental health, understanding its neural correlates is crucial (Saggar et al. 2015).

Previous neuroimaging studies of creativity have not consistently identified localized areas of the brain linked to this cognitive construct (Arden et al. 2010; Dietrich and Kanso 2010). This variation in findings may be partially explained by differing study designs as well as different definitions and operationalizations of creativity (Saggar et al. 2015). Another contributing factor may be that research on the neural correlates of creativity has largely focused on the concept of divergent thinking, which cannot fully encapsulate creativity as a whole (Dietrich and Kanso 2010). However, despite the large variance in results, there are some common findings. For example, the idea that the right side of the brain is linked to creativity, a popular cultural explanation of right- versus left-brain, is not backed up by any neurological research (Dietrich and Kanso 2010). Another common research finding is that the prefrontal cortices are implicated in creativity in one way or another (Dietrich and Kanso 2010). Some studies, however, report hyper-activation of prefrontal regions during creative thinking (Aziz-Zadeh et al. 2012) while others observe that deactivation of prefrontal regions facilitate creativity (Limb and Braun 2008; Saggar et al. 2015). Taken together, the current state of research suggests no single brain region is responsible for creativity and that creativity is instead a product of interactions between many areas within the brain (Arden et al. 2010; Dietrich and Kanso 2010).

To provide a more solid neuroscience foundation, researchers are now beginning to treat creativity as a multi-faceted construct and, as such, developing experiments that target individual aspects of creative thinking (e.g., "Aha" moment (Aziz-Zadeh et al. 2009), spontaneous improvisation (Pinho et al. 2014; Saggar et al. 2015), etc.). Better understanding of the brain basis of creativity also requires experimental paradigms that are both conducive to creative thinking and facilitate examination of applied creativity. Such experimental paradigms could help reduce variation in creativity neuroimaging results by minimizing confounding influences of cognitive processes that may be related to aspects of task design and have little or no relation with creative thinking (Saggar et al. 2015). For example, assessing creative thinking in a "test-like" setting as opposed to a fun/game-like environment is known to negatively influence creativity (Dietrich and Kanso 2010; Torrance 1987). Similarly, explicitly prompting participants to be "creative" during certain parts of the experiment and not creative during other parts is not only impractical but could also cause performance anxiety. Lastly, experimental paradigms should allow participants to freely express their creative potential in a relatively direct/unrestricted manner.

To address some of these methodological issues in studying the neural correlates of creativity, we recently developed a novel game-like and creativity conducive neuroimaging paradigm (Saggar et al. 2015). As creativity is a multifaceted construct, we specifically focused on one aspect of creativity, i.e., improvisation. Although improvisation is just one component of creative thinking, it can also encapsulate the prototypical creative behavior as a whole (Bengtsson et al. 2007). Free-form improvisation in musicians and artists have been previously used to study the neural correlates of enhanced creativity (Limb and Braun 2008; Pinho et al. 2014). Thus, we based our experimental paradigm on the social improvisation game of Pictionary™, where a player expresses a given word by sketching, while his/her teammates guess the guess the word by player's drawing alone. To capture the neural correlates of improvisation and figural creativity during this word-guessing task, we collected brain activity using functional Magnetic Resonance Imaging (fMRI) while participants were participating in the task.

In this chapter, we provide details regarding our approach, including insights and challenges faced during prototyping as well as a summary of our results to date. Towards the end of this chapter, building upon our novel paradigm, we provide suggestions for future work focused on assessing the neural correlates of creative capacity enhancement and team creativity.

2 Our Approach

To reduce the confounding influence of cognitive processes unrelated to creative thinking (e.g., anxiety or pressure), we endeavored to create a game-like, low-anxiety setting where participants can relatively freely express their creative potential. To create such a game like paradigm, where participants can spontaneously improvise solutions (to a given problem), we selected the well-known game of Pictionary™. This word-guessing game also provided us an opportunity to implicitly test creative potential of participants, rather than explicit prompting to be "creative" (as done in previous studies). To quantify creativity in these drawings we asked experts to later rate each drawing for creative content and ease of guessing the word.

Details of the implementation are provided elsewhere (Saggar et al. 2015). Briefly, the word-guessing fMRI game consisted of two conditions—(a) drawing the word; and (b) drawing zigzags. During both conditions participants used an MR-safe tablet (more details in next section) to draw their solutions. Participants were given 30 s for each condition in a block design. In the target condition (i.e., word-drawing condition), participants were instructed to draw the presented word (an action or verb) and told that other participants would be trying to guess what word he/she had represented by their drawing alone. In the second condition (i.e., zigzag-drawing), participants were asked to draw a representation for the control word “zigzag” in the same amount of time (i.e., 30 s). Thus, the second condition served as a control for the basic visuo-motor demands of the first condition. Hence, by subtracting brain activation during control condition from word-drawing condition we hypothesized to reveal the neural correlates of improvisation and figural creativity. Overall, the task lasted for 14.5 min and consisted of 10 blocks of each type of condition with a fixation period between blocks (duration: randomized between 10 and 15 s).

We used Matlab® and Psychtoolbox® software to implement the task in presentation hardware that interacted with the MR-scanner. Words presented during the word-drawing condition were selected from the “action word” grouping in Pictionary™. A separate group of participants (n = 10) rated the words in this grouping as either “easy,” “medium,” or “difficult” words to draw. Participants in the functional MRI study were presented three easy, four medium, and three difficult words in random (but fixed across subjects) order. After the scan, participants also rated how difficult the words had been for them to draw.

3 Challenges in Implementing Pictionary-Based Task

Creating and implementing such a task posed several challenges. Creating an experimental design that implicitly examines creative potential in non-artists/ musicians was difficult. No standard experimental paradigm exists to achieve these aims. In fact, most studies on creativity use a test-like setting with directions that, as mentioned above, can lead to a decrease in creativity and an increase in performance anxiety (Dietrich and Kanso 2010; Torrance 1987; Smith et al. 1990). Previous neuroimaging studies on creativity have also had incorporated many restrictions on participants with respect to demonstrating manifestations of creativity, such as by being limited to pushing buttons or “thinking” creatively without any “product” of such thinking.

We aimed to create an experimental paradigm that would allow observation of the neural correlates of creativity in an environment that fostered creativity without requiring explicit instructions to do so and in a manner that would put fewer constraints on participants’ creative expression. Key in making this task work was the development of an MR-safe drawing tablet for participants to sketch the representations of given words. Given that interacting with a tablet is quite common these days, we chose to develop an MR-safe drawing tablet (instead of other input devices, such as a mouse). It is important to note, that given the high magnetic strength of the MRI scanner, a regular tablet (with ferromagnetic components) could not have been allowed inside the scanner. As not many fMRI studies have used an MR-safe tablet before, we had to design and develop this tablet from scratch. Using a KEYTEC 4-wire resistive touch glass connected to a Teensy 2.0 electronic chip with custom firmware was implemented to create an

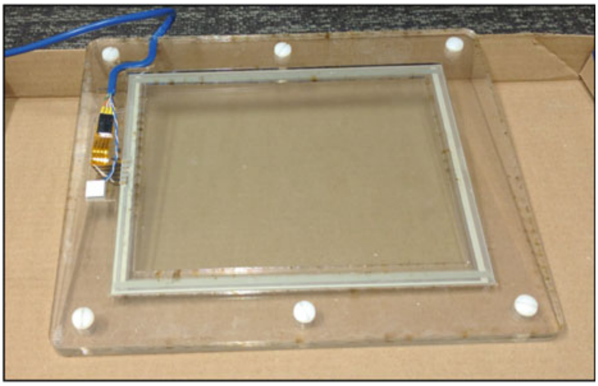


Fig. 1 MR-Safe drawing tablet, used by participants to draw the words while lying down in the MRI scanner

MR-compatible touch-sensitive surface (see Fig. 1). The tablet was connected to the presentation computer via a USB port. The tablet streamed absolute position of the pen (or finger) location using a simple serial protocol. The tablet case was build out of clear acrylic using a laser cutter. For future research purposes, we have freely released the tablet firmware online https://github.com/cni/widgets/tree/master/touch. Participants were trained to use the MR-safe drawing tablet before getting into the MRI scanner.

Functional MRI data acquisition is very sensitive to head movements (Power et al. 2012). Even small (~1–2 mm) head movements can create spurious activity in the blood-oxygen level dependent (BOLD) signal. One of the ways to reduce the spurious activation related to head movements is to regress out motion parameters from the data. However, if the head motion is correlated with task, regressing out motion can also regress out neural signal of interest. While using the MR-safe drawing tablet, participants also had head movements associated with drawing. Thus, it was challenging to remove spurious activity due to head movements that are associated with the drawing (or the task). To address this challenge, we employed spatial independent component analysis (spatial ICA). As the head movements were associated in time with the task, they were not necessarily associated in spatial location. Thus, for each participant, independent components were estimated and spatial components that were topographically deemed noisy (or movement related) were removed from the data. To label each component as an "artifact" or neural, as in-house semi-automatic artifact removal tool (SMART) was used (Saggar et al. 2012, 2015). SMART uses following rules to categorize each component as an artifact: (a) when the time series of a component is highly correlated ($r > 0.4$) with motion profile only and not at all with the task design; or (b) when a component has most of its power ($>70\,\%$) in the high frequency range. Once categorized, SMART produces an HTML based web-tool for quality check, where the operator can override SMART's automatic classification. After quality check the artifactual components were removed from the data to reconstruct artifact-free fMRI data for further analysis.

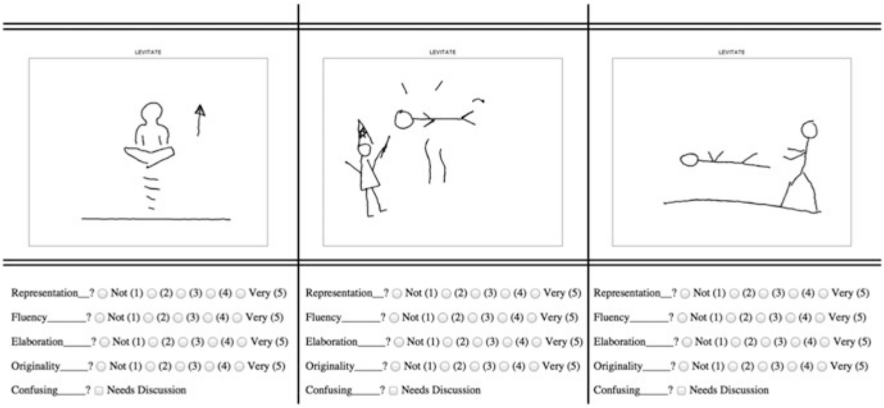


Fig. 2 A snapshot of the web-based interface developed for raters to score drawings on the representation scale and creativity sub-scales (fluency, elaboration, and originality). Example drawings shown here for the word—"levitate"

A third challenge in implementing this task was to efficiently rate creative content in the drawings produced during the fMRI task. A decision first had to be made on what aspects the drawings should be rated. Since rating drawings is inherently subjective, operational definitions for these aspects needed to be created. We decided to rate the drawings on two aspects—representation and creativity. To rate representation, raters considered the ease with which they believed another person could identify the word that was drawn. The expert raters then assigned a rating using a 5-point Likert scale where 5 indicated that a person could most easily guess the word upon which the drawing had been based. To obtain the creativity rating, 3 subdomains were selected and rated on a 5-point Likert scale. The average of these three ratings yielded the final creativity score. The subdomains were chosen from one of the standardized assessments of figural creativity (TTCT-F; Torrance 1990). The subdomains were fluency, elaboration, and originality. Fluency was defined as the number of unique components used to create the drawing. Elaboration referred to the quantity of creative detail, and originality referred to how uncommon a chosen representation was.

The raters then had to be able to reliably assess the drawings based upon the chosen criteria. Expert raters from the Stanford Design School (authors G.H. and A. R.) independently rated all the drawings. A web-based tool was created and de-identified drawing were shown to the raters through the website (Fig. 2). The expert raters trained on a sample consisting of 36 of the drawings. Based upon their Intraclass Correlation Coefficient, they achieved an inter-rater reliability of 0.80 on the representation domain and of 0.884 on the creativity domain across all drawings.

4 Results and Discussion

To identify the neural correlates of improvisation and figural creativity, we compared brain activation during the word drawing blocks versus the zigzag drawing blocks. For detailed results and discussion from the study, please see Saggar et al. (2015). Briefly, word- versus zigzag-drawing contrast revealed increased engagement of the cerebellum, thalamus, left parietal cortex, right superior frontal, left prefrontal and paracingulate/cingulate regions. Further, higher activation in the cingulate and prefrontal regions was associated to lower expert representation rating scores. Also, unexpectedly, we observed increased engagement of bilateral cerebellum with increasing expert creativity rating. As cerebellar-cerebral interactions are previously implicated in implicit processing, our data suggested a direct role of implicit processing via cerebral-cerebellar interaction during improvisation and figural creativity.

To better understand these neuroimaging results, we used Ito's (2008) model of problem solving as a starting framework (Ito 2008). Ito's model includes both explicit processing (through working-memory and attentional systems) as well as implicit processing (via cerebral-cerebellar processing). Previous theoretical research on creative thinking has suggested extending Ito's model to understand creative thought process (Ito 2008; Vandervert et al. 2007). For example, when employing divergent thinking tasks to measure creative thinking, the model predicts explicit processing to be activated as the participants try to generate alternate, novel and unique solutions to open-ended problems. Similarly, for an intuitive leap or an Aha moment to happen, the model predicts engagement of implicit processing via cerebral-cerebellar interactions, where the "Aha" moment happens when the solution reaches conscious awareness.

The fMRI task used in this study required participants to consider existing mental representations for different concepts and to modify them in order to produce a figural representation of the given word. Recruitment of the cerebellum may have allowed engagement of implicit processing to synthesize and improvise representations of a given word efficiently (Saggar et al. 2015). It remains unclear, however, how such greater efficiency is translated to creativity. Future research paradigms are required to systematically dissociate the cerebellum's role in different aspects of creative thinking (e.g., elaboration, flexibility, etc.).

Future research should also continue to use cognitive models as a framework to better understand the neural correlates of creativity as a whole. A potential limitation of the design used for our study was the use of the zigzag blocks to control for non-target brain activity. Further control scenarios should be tested to see whether the zigzag condition in our study adequately controlled for the non-creative brain processes that our word-drawing condition entails. Future studies could use more a cognitively demanding control block, such as tracing a path through a maze. Another limitation of this study may have been the presentation of a relatively small number of word-drawing blocks to the participants, resulting in an insufficient number of drawings with which to assess the relationship between word difficulty, drawing representation and creativity. Our study also lacked the ability to distinguish between the pre-drawing planning and the actual drawing of the figures. Future studies can incorporate a "planning" block in which the participant is asked to spend a block of time planning their drawing before they actually draw the figure on the tablet.

5 Future Directions and Impact

Based on our work in developing a novel game-like creativity assessment tool, we plan to employ this assessment tool to assess the neural correlates of creative capacity enhancement and sustainability. Further, to examine neural correlates of team creativity, we plan to extend our word-guessing task from individual settings to team play. The role of the cerebellum in creativity also requires further investigation. In near future, we have planned a study to determine whether portable neuroimaging techniques (e.g., near-infrared spectroscopy; NIRS) can be used to accurately monitor activity in the cerebellum. The ability to use NIRS would allow for a wider range of tasks to assess creativity in more naturalistic settings and with a larger numbers of interacting individuals.

References

Abraham A (2013) The promises and perils of the neuroscience of creativity. Front Hum Neurosci 7:246. doi:10.3389/fnhum.2013.00246

Arden R, Chavez RS, Grazioplene R, Jung RE (2010) Neuroimaging creativity: a psychometric view. Behav Brain Res 214(2):143–156. doi:10.1016/j.bbr.2010.05.015

Aziz-Zadeh L, Kaplan JT, Iacoboni M (2009) "Aha!": the neural correlates of verbal insight solutions. Hum Brain Mapp 30(3):908–916. doi:10.1002/hbm.20554

Aziz-Zadeh L, Liew S-L, Dandekar F (2012) Exploring the neural correlates of visual creativity. Soc Cogn Affect Neurosci 8(4):475–480. doi:10.1093/scan/nss021

Bengtsson SL, Csíkszentmihályi M, Ullén F (2007) Cortical regions involved in the generation of musical structures during improvisation in pianists. J Cogn Neurosci 19(5):830–842. doi:10.1162/jocn.2007.19.5.830

Dietrich A, Kanso R (2010) A review of EEG, ERP, and neuroimaging studies of creativity and insight. Psychol Bull 136(5):822–848. doi:10.1037/a0019749

Ellamil M, Dobson C, Beeman M, Christoff K (2012) Evaluative and generative modes of thought during the creative process. NeuroImage 59(2):1783–1794. doi:10.1016/j.neuroimage.2011.08.008

Ito M (2008) Control of mental activities by internal models in the cerebellum. Nat Rev Neurosci 9 (4):304–313. doi:10.1038/nrn2332

Limb CJ, Braun AR (2008) Neural substrates of spontaneous musical performance: an FMRI study of jazz improvisation. PLoS One 3(2), e1679. doi:10.1371/journal.pone.0001679

Pinho AL, de Manzano Ö, Fransson P, Eriksson H, Ullén F (2014) Connecting to create: expertise in musical improvisation is associated with increased functional connectivity between premotor and prefrontal areas. J Neurosci 34(18):6156–6163. doi:10.1523/JNEUROSCI.4769-13.2014

Power JD, Barnes KA, Snyder AZ, Schlaggar BL, Petersen SE (2012) Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. NeuroImage 59(3):2142–2154. doi:10.1016/j.neuroimage.2011.10.018

Saggar M, King BG, Zanesco AP, MacLean KA, Aichele SR, Jacobs TL et al (2012) Intensive training induces longitudinal changes in meditation state-related EEG oscillatory activity. Front Hum Neurosci 6:256. doi:10.3389/fnhum.2012.00256

Saggar M, Quintin E-M, Kienitz E, Bott NT, Sun Z, Hong W-C et al (2015) Pictionary-based fMRI paradigm to study the neural correlates of spontaneous improvisation and figural creativity. Sci Rep 5:10894. doi:10.1038/srep10894

Smith KLR, Michael WB, Hocevar D (1990) Performance on creativity measures with examination-taking instructions intended to induce high or low levels of test anxiety. Creativity Research Journal 3(4):265–280, http://doi.org/10.1080/10400419009534360

Srivastava S, Ketter TA (2010) The link between bipolar disorders and creativity: evidence from personality and temperament studies. Curr Psychiatry Rep 12(6):522–530. doi:10.1007/s11920-010-0159-x

Torrance EP (1968) A longitudinal examination of the fourth grade slump in creativity. Gifted Child Q 12(4):195–199

Torrance EP (1987) Guidelines for administration and scor ing/comments on using the torrance tests of creative thinking. Scholastic Testing Service, Bensonville, IL

Torrance EP (1990) Torrance tests of creative thinking: figural forms A and B. Scholastic Testing Service, Bensonville, IL

Vandervert LR, Schimpf PH, Liu H (2007) How working memory and the cerebellum collaborate to produce creativity and innovation. Creativity Res J 19(1):1–18

Erratum: Design Thinking Research

Hasso Plattner
Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany
hasso.plattner@hpi.unipotsdam.de

Christoph Meinel
Hasso Plattner Institute for Software Systems Engineering
Potsdam, Germany
christoph.meinel@hpi.unipotsdam.de

Larry Leifer
Stanford University
Stanford, CA, USA
leifer@cdr.stanford.edu

H. Plattner et al. (eds.), *Design Thinking Research*, Understanding Innovation

---

DOI 10.1007/978-3-319-40382-3_18

Errata to:

Chapter 6: On Creating Workspaces for a Team of Teams: Learnings from a Case Study, doi: 10.1007/978-3-319-40382-3_6

Marie Klooker, Claudia Nicolai, Stephan Matzdorf, Arne Trost, Karen von Schmieden, Lilith Böttcher and Ulrich Weinberg

Chapter 16: The Topic Markup Scheme and the Knowledge Handling Notation: Complementary Instruments to Measure Knowledge Creation in Design Conversations, doi: 10.1007/978-3-319-40382-3_16

Axel Menning, Andrea Scheer, Claudia Nicolai and Ulrich Weinberg

In the original version for these papers, the name of the author Ulrich Weinberg was omitted.

The updated original online version for these chapters can be found at
http://dx.doi.org/10.1007/978-3-319-40382-3_6
http://dx.doi.org/10.1007/978-3-319-40382-3_16