

InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup



Franziska Dobrigkeit, Danielly de Paula, and Matthias Uflacker

Abstract The debate on how to integrate Design Thinking and Lean Startup into the agile process has been addressed in the literature over the past few years. Researchers argue that Design Thinking can contribute to software development by offering support on how to understand user needs in order to derive solution and product options, whereas Lean Startup helps to learn about business and scaling strategies. Based on these viewpoints, we developed InnoDev, which is an approach that combines Design Thinking, Lean Startup and Scrum to create an agile software development process that can deliver the innovative customer-oriented products and services required by competitive companies. This study aims to describe InnoDev in detail by depicting all its phases. Our findings provide complementary perspectives regarding software development strategies, roles and techniques. This study will advance the knowledge of Design Thinking and software development by providing a detailed description of a tool that combines best practices for creating more innovative software products. The results of this investigation can help managers to evaluate their software development process in order to improve its effectiveness and create more efficient user-driven solutions.

1 Introduction

Over the years, many scholars have emphasized the importance of design for software development. For instance, Frishammar & Florén found that the early design or concept creation phases in process development are critical to the final results, and boast a large potential for cost savings (Frishammar et al. 2011). Additionally, Chin et al. focus on how to make better screening decisions for new product ideas based on the customer needs (Chin et al. 2008). Researchers suggest the integration of Design Thinking with agile software development—particularly,

F. Dobrigkeit (✉) · D. de Paula · M. Uflacker
Hasso Plattner Institute for Digital Engineering, Potsdam, Germany
e-mail: franziska.dobrigkeit@hpi.de; danielly.dePaula@hpi.de; matthias.uflacker@hpi.de

in order to improve problem understanding and solutions, as well as improving attention towards design (Lindberg et al. 2011).

Agile methods (e.g. Kanban, Scrum) have been recommended for software development due to their benefits in relation to reducing the development time, and increasing the flexibility and quality of the product (Erickson et al. 2005). However, a comprehensive systematic literature review conducted by Dybå and Dingsøyrr concluded that a limitation which has repeatedly been mentioned in the literature related to Scrum is the lack of attention to design (Dingsøyrr et al. 2012). These limitations can lead to severe consequences such as: a company launching the “wrong” products, resulting in poor market reception, or necessary rework requiring extra engineering hours and investments (Verganti 1997). In order to overcome these restrictions and encourage more interdisciplinary collaboration, there have been serious efforts to introduce design methods, especially Design Thinking, to IT development (Hildenbrand and Meyer 2012; Hirschfeld et al. 2011; Lindberg et al. 2011).

The debate of how to integrate Design Thinking into the agile process has been addressed in the literature in the past few years. For instance, Grossman-Kahn and Rosensweig suggest that software development teams should be guided by a clearly defined set of end goals and mindsets such as Design Thinking, agile and lean (Grossman-Kahn and Rosensweig 2012). Similarly, Hildenbrand and Meyer introduced the concept of lean thinking and developed a model using Design Thinking and agile to optimize the training experience for software professionals and their coaches (Hildenbrand and Meyer 2012). By combining the models of Grossman-Kahn and Rosensweig (2012) and Hildenbrand and Meyer (2012), de Paula developed a new model that aims to identify, implement and scale solutions in a startup environment (de Paula 2015). Häger et al. present DT@Scrum, a process model for large organizations that seamlessly integrates Design Thinking and Scrum (Häger et al. 2015).

Based on the mentioned studies, we developed InnoDev. InnoDev is an approach that combines Design Thinking, Lean Startup and Scrum in order to create an agile software development process that can deliver the innovative customer-oriented products and services required by competitive companies. This study aims to describe InnoDev in detail by depicting all its phases. This study will advance the knowledge of Design Thinking and software development by providing a detailed description of a tool that combines best practices for creating more innovative software-products. The results of this investigation can help managers to evaluate their software development process in order to improve its effectiveness and create more efficient user-driven solutions.

We consider InnoDev to be a general model applicable to different company settings (e.g. Startups, SMEs, or large organizations), however; it is necessary to validate the model with companies in order to identify whether our claim makes sense. Therefore, future research will evaluate InnoDev through an objective and systematic validation process using a combination of workshops designed to teach InnoDev and a survey designed to validate InnoDev with a large sample of software development companies.

The remainder of this article is structured as follows: Section 2 provides an overview of existing research on Design Thinking, agile and Lean Startup for software development. Our research approach is described in Sect. 3 and our general findings around InnoDev are presented in Sect. 4. Section 5 presents our evaluation protocol, and Sect. 6 closes this article with a summary.

2 Related Work

Software development has been using agile methods, such as Kanban or Scrum for several years now. These methods were developed and became popular because they often reduce the development time and increase the flexibility and quality of the product (Erickson et al. 2005). The most common approach to agile software development is Scrum (Komus 2017; Scrum Alliance 2016; Version One 2017). Scrum focuses especially on project management for projects and situations that are difficult to plan in advance, by introducing feedback loops, self-organizing teams and 1–4 week sprints as core elements (Schwaber and Beedle 2001). Another popular methodology is Design Thinking. Design Thinking has also been around for several years now and has shown to be successful as a way to develop superior products and to facilitate product appropriateness by enhancing team collaboration and improving idea generation (Beverland and Farrelly 2007). At the core of Design Thinking are four key elements: the iterative process, including various methods and tools supporting each phase; multidisciplinary teams; creative space; and a designer’s mindset (Wölbling et al. 2012).

Nevertheless, both of these methods are not without shortcomings. Dybå and Dingsøy reported a weakness repeatedly found during their comprehensive systematic literature review to be a lack of attention to design in Scrum projects (Dybå and Dingsøy 2008). Similarly, Lindberg et al discuss that understanding the customer and finding the right solution are common among IT teams, especially in agile teams (Lindberg et al. 2011). On the other hand, Design Thinking is often criticized for not looking at the actual implementation or production of the ideas generated (Wölbling et al. 2012). Additionally, neither Design Thinking nor agile practices offer support on how to track growth and how to scale a product after its launch (Grossman-Kahn and Rosensweig 2012; Vilkki 2010).

Introducing design methods and Design Thinking to agile IT development teams is a solution proposed in literature as a way of overcoming some limitations and encouraging more interdisciplinary collaboration (Hildenbrand and Meyer 2012; Lindberg et al. 2011). As agile methods often fail to describe how requirements are gathered before the actual development, these researchers propose Design Thinking as a pre-phase to development, aimed at analyzing and eliciting requirements. This approach promises cost savings due to reductions in redesign work, as well as shortening the length of the process itself (Lindberg et al. 2011). Additionally, several authors propose including Lean Startup in such a combined methodology in order to address the issues of scaling and tracking growth (Grossman-Kahn

Table 1 Processes combining Design Thinking-and agile software development

Model	Specialty	Focus	Target group
Grossman-Kahn and Rosensweig (2012)	Lean Startup integrated and tested in a laboratory	Identify the solution + deliver a prototype	Startups
Hildenbrand and Meyer (2012)	Using lean thinking concepts throughout the development process	Identify + implement the solution	Inexperienced teams
Müller and Thoring (2012)	Combining Lean Startup and Design Thinking	Implement + scale the solution	Entrepreneurs, innovators, and startups
Häger et al. (2015)	Using Scrum to structure Design Thinking activities	Identify + implement the solution	Large software organizations
de Paula and Araújo (2016)	Integrating Lean Startup into a startup environment and testing it with students	Identify, implement + scale the solution	Startups

and Rosensweig 2012; Hildenbrand and Meyer 2012; de Paula and Araújo 2016). Lean Startup, with its build-measure-learn-lifecycle, aims at providing guidance on how to develop a product that meets its value proposition in an MVP—a Minimum Viable Product without waste (Ries 2011). Additionally, Lean Startup includes actionable metrics to assess the product performance and the user’s acceptance (Maurya 2012), the business model canvas or lean canvas to develop the business side of a product and the concept of a pivot, “a special kind of change designed to test a new fundamental hypothesis about the product, business model, and engine of growth” (Ries 2011).

The characterization of new theories on how to integrate Design Thinking into the agile process has been progressing in the literature. Table 1 summarizes a selection of existing models. Grossman-Kahn and Rosensweig propose a design-led, multidisciplinary model to build innovation capacity through the integration of diverse innovation methodologies such as Design Thinking, Lean Startup and agile practices (Grossman-Kahn and Rosensweig 2012). By validating the model with a team from the Nordstrom Innovation Lab, the authors suggest that software development teams should be guided by a clearly defined set of end goals and mindsets, rather than a rigid adherence to specific tools or processes. Similarly, Hildenbrand and Meyer 2012 introduced the concept of lean thinking and developed a model using Design Thinking and agile methods to optimize the training experience for software professionals and their coaches (Hildenbrand and Meyer 2012). The authors suggest that lean thinking is closely intertwined with Design Thinking in many ways and they complement each other very well. Müller and Thoring compare Design Thinking and Lean Startup in detail, highlighting gaps, differences and intersections between the two innovation strategies (Müller and Thoring 2012). They believe Design Thinking and Lean Startup can benefit from each other since they each have features the respective other methodology is missing. As a cumulation of these thoughts they propose Lean Design Thinking, a methodology merging Design Thinking and Lean Startup. Häger et al. and Vetterli et al. present DT@Scrum, a process model for large organizations that seamlessly integrates

Design Thinking and Scrum (Häger et al. 2015; Vetterli et al. 2013). Unlike the other models, the authors use agile concepts, such as sprints and backlogs, to plan and structure the Design Thinking activities. de Paula and Araújo (de Paula 2015; de Paula and Araújo 2016) developed a model using agile, Lean Startup and Design Thinking by combining the models of Grossman-Kahn and Rosensweig (2012) and Hildenbrand and Meyer (2012). It is based on previous research (de Paula et al. 2014) and aims to identify, implement and scale solutions in a startup environment.

Although several processes and concepts that combine Design Thinking, agile practices and Lean Startup already exist, a generally accepted model has not yet emerged. Building upon the latest concepts, DT@Scrum and MOIT (which in turn were created based on some of the former concepts) de Paula and Dobrigkeit developed InnoDev (Dobrigkeit and de Paula 2017), which will be described in more detail in Sect. 4.

3 Method

In the following sections, we aim to describe each element of InnoDev in detail. To do so we will use elements that are common to method descriptions as collected by Gutzwiller. He derived five key elements as part of a method description: activities, roles, deliverables, techniques and the meta model (Gutzwiller 2013). An activity in this context describes a unit that aims to produce one or more defined results. Such activities can be structured hierarchically or in sequence. Activities are run by people or a group of people in certain roles. Roles in such cases describe a combination of activities from the view of the actor. Deliverables are the results of activities. They can also function as an input to activities and can thereby either be created or modified during activities. Techniques are tools or methods that support the creation of the deliverables. Compared to activities they are more detailed and on a smaller level. The meta model is the conceptual data model of the deliverables. The five elements and their relationships are depicted in Fig. 1.

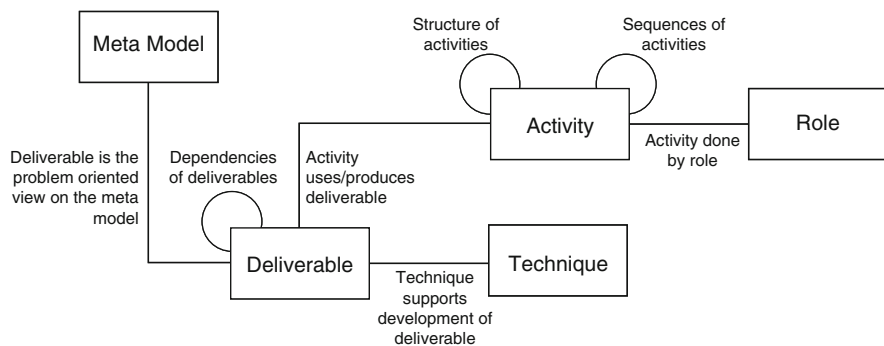


Fig. 1 Elements of a method description (translated from Gutzwiller, 1994, p. 13)

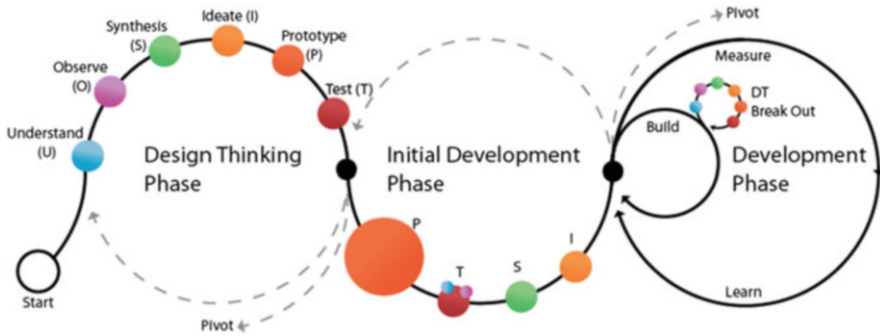


Fig. 2 InnoDev Process

4 InnoDev in Detail

Similar to former process proposals, InnoDev is based on three phases: The Design Thinking phase, the Initial Development phase and the Development phase. The process and its phases are depicted in Fig. 2.

The main difference between the three phases is the ratio between Design Thinking, Lean Startup and development activities. With an increasing understanding of the problem and the requirements for a solution, the team decreases Design Thinking activities and increases software development and business building. Lean Startup and Scrum concepts are present during all phases: each phase can be seen and implemented as a build-measure-learn-lifecycle, making use of the sprint and backlog concepts from Scrum to plan and structure the necessary activities. Thus, transparency in all activities is provided alongside flexibility to move forward with constant learning even with changing requirements or pivoting if necessary.

Before starting in the Design Thinking phase, a challenge or a general area of interest should be available to the InnoDev Team. Such a statement could be defined by the team itself or be issued by a manager or someone else outside the team. It can come in the form of a problem statement, a design brief or a simple sentence but should give the team a broad idea of the subject matter to investigate during the Design Thinking phase. Additionally, it is helpful if the team has access to potential users and stakeholders from the beginning and is sufficiently trained to use Design Thinking, Scrum and Lean Startup techniques. Armed with these pre-requisites the InnoDev Team can start the first phase of the process.

The Design Thinking phase emphasizes Design Thinking activities and is aimed at understanding user needs and related products. This phase follows the Design Thinking process as described by Wölbling et al. (2012) and Thoring and Müller (2011) to explore the problem and solution spaces and define a product vision addressing at least one of the identified problems. During this phase, Lean Startup activities support the validation of early ideas with metric-based testing and Scrum practices support project planning.

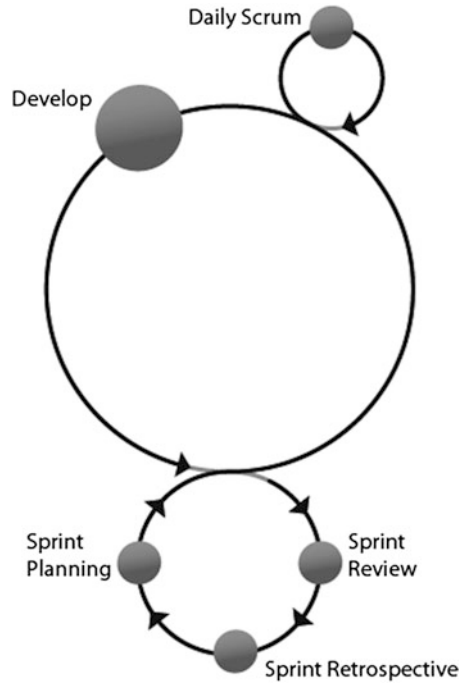
The Initial Development phase aims to refine and test the product vision from the Design Thinking phase with respect to desirability, technical feasibility and business viability ultimately arriving at a proof-of-concept prototype, following the idea of an MVP. This phase balances activities from Design Thinking, Lean Startup and software development. Business models around the concept are created and validated and ways of collecting data to monitor user acceptance are implemented. On the development side, UI as well as technical concepts are created and tested and the most important features are implemented. Throughout this phase, Design Thinking activities help to prototype, test, and refine the product vision as well as the business model and the technical concepts. Project management is done using Scrum.

In the final Development phase, the MVP will be tested and gradually extended into a full featured product according to the original concept or feedback gained during this or the previous phase. The business model and technology architecture are scaled accordingly. In this phase, the team will run agile sprints combined with lean practices to establish a build-measure-learn-lifecycle. Depending on the outcome of the learn phase the team decides whether to pivot their project or continue to the next sprint. While this phase is focused on development and scaling, InnoDev proposes to make use of Design Thinking tools in an ad-hoc manner in case of blockers or problems related to either the product or the process.

4.1 Scrum: The Overall Project Management Method Underlying All Phases of InnoDev

To structure the work during all phases of InnoDev, Scrum project management tools provide an overall framework. Scrum proposes planning work in smaller cycles of 1–4 weeks, a so-called sprint (Deemer et al. 2012; Schwaber 1997; Schwaber and Sutherland 2013). Each sprint consists of a planning, working, and reflecting on the work done and the deliverables created. All three methods that are merged in InnoDev are essentially centered around trying and learning, each using different tools and techniques. Design Thinking aims to understand and learn about problems and user needs in order to derive solutions and product options. Lean Startup aims to learn about business strategies and scaling options and Scrum aims to learn about development options and further directions during software development with changing requirements. As such, each phase of InnoDev can be considered as a build-measure-learn-lifecycle as presented by Ries (Eisenmann et al. 2012; Ries 2011)—albeit each with a different focus and therefore with a different set of actions, deliverables techniques, and roles used. Consequently, techniques to reflect on the achieved work and adjust the project accordingly are also inherent to each of the methods. In this way, Scrum is a useful method to streamline these techniques into an overall project management framework. Figure 3 depicts the core activity of the Scrum framework the sprint.

Fig. 3 Basic Scrum process



4.1.1 Activities

The core activity of the Scrum framework is a time-boxed development sprint at whose end a working version of the system under development is created. Such a sprint allows an ongoing validation of the product with the customer requirements and thereby allows the alteration of requirements if necessary. The Scrum framework proposes four meetings within one sprint: a sprint planning meeting, to decide on the work for the upcoming sprint, a review meeting, to reflect on the produced deliverables, a retrospective meeting, to reflect on the team work and process, and a short stand-up meeting, to discuss progress open work and issues in the team during each sprint. The sprint planning, the review and the retrospective should each occur once every sprint while the stand-up meeting should occur daily at the beginning of the workday.

In the context of InnoDev not all of these meetings are relevant for all phases. The sprint planning meeting and the retrospective should be held throughout the whole InnoDev process. However, retrospectives might not be necessary after every sprint. Especially during the Design Thinking phase retrospectives of the team work and process can be found as a short daily check-out. Therefore, not every sprint requires a retrospective meeting and they can be held if necessary during this phase. The review meeting is only relevant when work is split between team members. However, when working in the Design Thinking phase a lot of work is done with the

whole team, thus working in sub-teams requires an immediate review of the deliverables produced. Therefore, the review meeting is unnecessary during this phase. The daily stand-up meeting on the other hand already exists in some implementations of Design Thinking as a so-called team check-in. This meeting is also known to Lean Startup experts, as they use agile development practices.

4.1.2 Roles

Scrum proposes three main roles, the Product Owner, the Scrum Master, and the Development Team. The Product Owners is responsible for collecting requirements from users and other stakeholders of the system. He also transfers them into small, understandable and implementable pieces (often in the form of agile user stories).

The Scrum Master is a specially trained moderator and coach, supporting the product owner and the development team by moderating the meetings and solving problems that occur along the way. He also makes sure everyone adheres to the rules of Scrum.

The Development Team takes care of the work items planned for each sprint and implements the selected requirements.

Again, not all roles are relevant to each phase of InnoDev. The role of the Scrum Master makes sense throughout the InnoDev process. However, as several methods and techniques play a role in InnoDev this role could be merged with supporting roles from the other methodologies such as a Design Thinking coach. The role of the product owner only makes sense once a product vision exists. Before that each member of the team should partake in eliciting and prioritizing requirements. Because in the earlier phases of InnoDev “development team” might be a misleading title, we will use only the term “team” or “InnoDev Team” instead.

4.1.3 Deliverables

The main deliverables of the Scrum framework are the Product Backlog—a collection of work items necessary to complete the project, the Sprint Backlog—the collection of work items due during the current sprint and the working software increment—the outcome of a sprint.

Scrum was developed to create software and therefore the wording of the deliverables matches this context. For InnoDev, we propose a more general wording that fits for all three phases of our process. The Product Backlog becomes the Project Backlog which will be filled with Design Thinking tasks in the beginning and only later will include software requirements. The name Sprint Backlog is still valid; however, the contents of this backlog change according to the Project Backlog. The working software increment simply becomes an increment in InnoDev. It can describe any form of progress such as user needs or solution concepts during the early phase of InnoDev or actual software prototypes and working software in the later phases.

4.1.4 Techniques

Scrum itself does not propose specific techniques. However, a number of techniques to support Scrum activities have been developed and reported. Interesting for InnoDev are the collections of retrospective games by Kerth, Derby et al., and Kua (Derby et al. 2006; Kerth 2000; Kua 2013) (which are similar to various Design Thinking techniques) and planning techniques, such as planning poker or bucket planning, which allow for quick planning in a team (Grenning 2002). Furthermore, the use of a task or Scrum board makes sense in tracking the current activities and progress in an easy and flexible way.

4.2 Design Thinking Phase

The Design Thinking phase, as depicted in Fig. 4, mainly uses Design Thinking techniques to find and explore the projects' problem statement and the solution space. Additionally, Design Thinking and Lean Startup techniques are used to validate first solution concepts. The goal of this phase is to come up with a clear product vision and corresponding low-resolution prototypes and user stories.

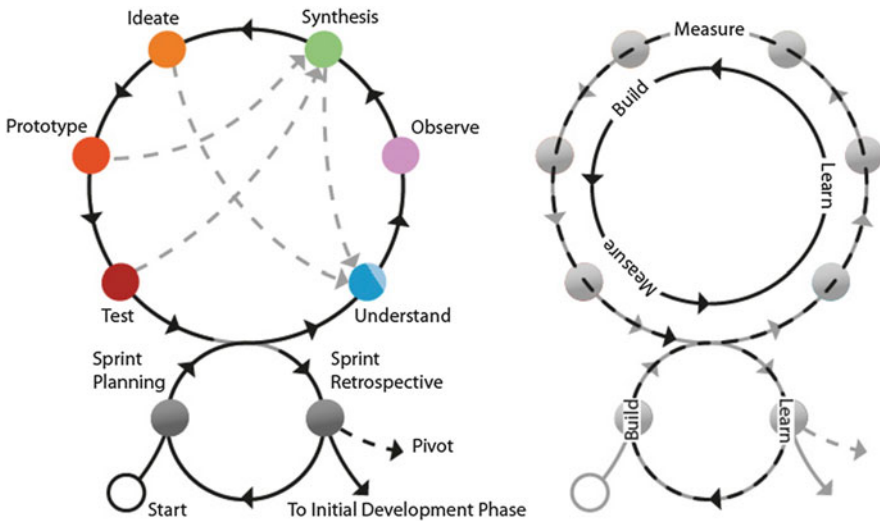


Fig. 4 Overview of the Design Thinking phase and representation of the build measure and learn concepts within this phase

4.2.1 Activities

The main activity during this mode is the Design Thinking process as described by Wölbling et al. (2012) or Thoring and Müller (2011). The team starts out with a general problem statement or an area of research and uses the initial Understand phase to collect information about the projects goals, constraints, and environment. In the following Observe phase, the problem domain is further investigated by looking at existing solutions, and getting in contact with real users and stakeholders. In the Synthesis or Point of View phase, the team reviews all the information gained from the first two phases and aims to condense them into their Point of View on the problem. In the following phase, the team uses the condensed problem statement as a basis for ideating possible solution ideas. The team reviews the generated ideas and selects promising candidates to prototype during the prototyping phase. The prototypes thus created will undergo qualitative and metric-based testing with end-users. The results of these tests will be synthesized again and, depending on the outcome of that synthesis, the team can decide to either iterate on the solution to refine it, continue with other solution ideas, or pivot and go back to “understanding” and “observing” to get a new view of the problem.

4.2.2 Techniques

Design Thinking and Lean Startup and Scrum each come with their own sets of techniques, which are useful in understanding the project environment, stakeholders, users, problem space, and solution space. Useful techniques from the Design Thinking toolbox include: 360° research, observation and interview techniques, storytelling, synthesis techniques, brainstorming techniques, various forms of prototypes, and testing techniques. This set is complimented by metric-based measurement techniques from the Lean Startup toolbox and planning techniques from the Scrum toolbox. In the following, we give a short description of these techniques and their purpose. As the number of techniques is large and still growing, the following descriptions should be considered examples and not the only usable techniques during this phase.

360° research is essentially a desk and internet research, which allows the team to quickly become well-versed in a new topic. Observation and interview techniques enable the team to get in contact with stakeholders and end-users to understand their views on the topic and problem. They can also get an understanding of their needs and pains. These techniques include: shadowing, observing participants, interviewing groups and individuals, and seeking out extreme users (d.school Stanford, n.d.). Storytelling is a great technique to share information gathered during interviews, observation or testing within the team (d.school Stanford, n.d.). Synthesis techniques aim to capture information gathered by the team and arrange it in a form that provides an overview or organizes it in a way that makes it possible to convey or derive new findings. Examples of such tools are Stakeholder maps

(Freeman 2010), Personas, Point of View Statements, a 2-by-2 Matrix or Venn-Diagrams. Brainstorming techniques are used to generate solution ideas for discovered needs or personas. They include techniques such as, hot potato brainstorming, silent brainstorming or body storming.

Prototyping techniques can be used to understand the user and the problem statement or to validate ideas. During this phase prototypes will mostly be rough and quick and easy to build for example cardboard or paper prototypes, sketches of user interfaces, or even role plays. Naturally such prototypes need to be tested with actual users to get feedback on the current solution idea and discover flaws and further needs. Testing techniques include observing users while they are trying everything out and then interviewing them afterwards using testing protocols. Furthermore, landing pages are a good way to test the user's interest in an idea. This is done by creating a website describing the future product and either tracking how and when people find the page, or adding a subscription form to actually see how many people subscribe.

4.2.3 Roles

The InnoDev Team is responsible for planning and executing the sprints during the Design Thinking phase. Such a team should consist of people from different areas of expertise, e.g. accounting, sales people, UI designers, developers or consultants, depending on what knowledge will be relevant for the project.

Potential users and stakeholders provide insights on the topic and their problems and give feedback on ideas prototypes and the project in general. For that purpose, potential users are interviewed and observed by the InnoDev Team. Potential users originate from a broad range of users in the beginning of this phase and then gradually narrow down to a target user group. The InnoDev Team tries to secure people from this group for continuous testing and feedback cycles.

For most projects, a person or group of people has formulated the original challenge. These people can be external customers or partners, or internal project sponsors (e.g., customer representatives or managers). Either way the people in this role, whom we call project sponsors, are the first contacts the team has as a way to reaching experts dealing with their challenge or topic. Thus, the person in this role is responsible for providing initial material (e.g., reports on former projects or products, market or other research that was already done or a general introduction to the topic of interest), and initial interview partners (e.g., knowledge experts inside the company or possible customers). Additionally, those in this role connect the team with other departments inside the company to enable synergistic effects and avoid duplicate efforts. Furthermore, who serves in this role provides feedback in the same way potential or target users do.

The InnoDev Facilitator can be one or more people who support the InnoDev Team. They help the team navigate through the process by introducing useful techniques, helping each role to understand what to do and how to do it, and helping the team to solve problems that arise along the way. Additionally, whoever serves in

this role is responsible for moderating team meetings and discussion, as well as watching team dynamics. As such, the person or people in this role should have a solid understanding of InnoDev and its components Design Thinking, Lean Startup and Scrum in order to provide useful techniques and guidance at proper times.

4.2.4 Deliverables

The main output of this phase is a clear product vision, which will be further tested and refined in the Initial Development phase. Along the way in this phase the InnoDev Team will produce knowledge that should be documented in quick and easy ways, thus making it possible to trace ideas and decision. Such lightweight documentation could include interview summaries, collecting the main insights; filled out synthesis frameworks such as personas or matrices and diagrams; idea sheets documenting the core concepts of promising ideas, various low-resolution prototypes as well as one or more sophisticated solution prototype. These materials should make clear why each aspect of the product vision and the solution prototype have been designed the way they have. In order to make the step into the Initial Development phase, the InnoDev Team will create high-level user stories and a list of non-functional requirements based upon the materials created, the product vision and the solution prototype.

4.3 Initial Development Phase

The main goal of the Initial Development phase as shown in Fig. 5 is to create a minimum viable product based on the solution prototype and the product vision created in the Design Thinking phase. To that end, the solution will be further refined with a special focus on ensuring viability, feasibility and desirability through further exploring and testing not only the solution itself but also of possible business models and technologies to use for implementation. The outcomes of this phase will be higher resolution prototypes of the solution, the MVP, refined user stories, a list of non-functional and technical requirements and a business model.

4.3.1 Activities

The main activities of this mode are the further refinement of the solution, the validation of technical aspects and the technology to use, the creation and validation of a UX Design, the creation and validation of a business model, and the implementation of a minimum viable product, a working software system, albeit only with the most essential features and not necessarily on the final technology stack or in the final design. As such, this phase focuses on various prototyping and testing activities in the areas of software development, UX and Design and business development.

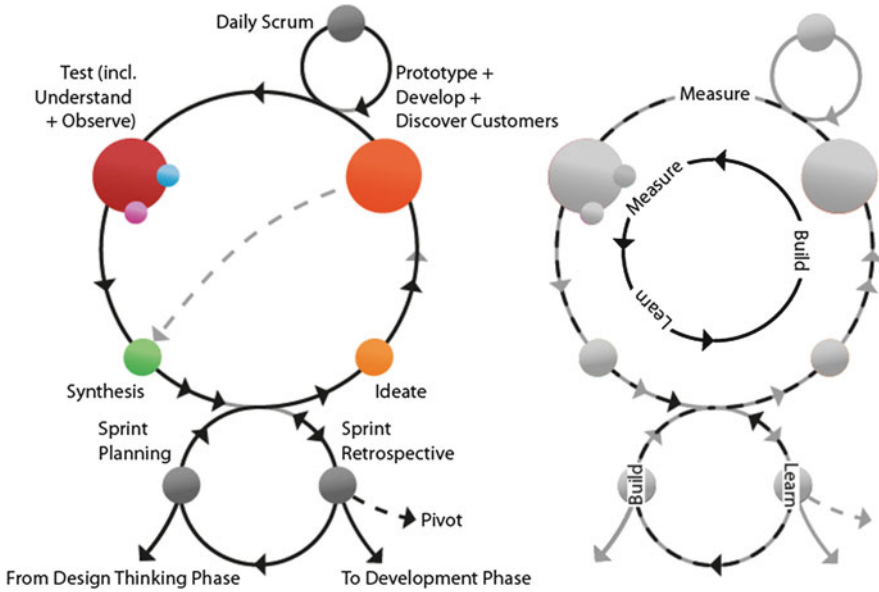


Fig. 5 Overview of the Initial Development phase and representation of the build measure and learn concepts within this phase

Initially the InnoDev Team identifies aspects of the solution prototype and the product vision that need further clarification, either in terms of detailing the concept, ensuring technical feasibility or business viability. These aspects are then refined by running through steps of the Design Thinking process as necessary (e.g., if it is unclear whether other features are needed, further interviews and observations can be initialized; if different concepts for a feature exist, these can be prototyped and tested with target users, if new features should be added, further ideation will help; or in case technical feasibility is unclear a proof-of-concept prototype can be developed. After the first refinements, and parallel to further refinements, a UI design, a software architecture, and the business model will be created and validated in further testings. Additionally, technology options are being evaluated. The knowledge gained through all these activities is then used to decide on an MVP and implement it along with further refinement of the business model and the UI and software design. Additionally, the existing user stories and requirements lists should be updated according to the new knowledge leading to more refined user stories as well as non-functional and technical requirements

4.3.2 Techniques

Core techniques during the Initial Development phase are again taken from the toolboxes of each original methodology. Design Thinking techniques used during this phase include mid-fidelity prototyping techniques and qualitative testing

techniques. Lean Startup techniques include business model creation, further measurement based testings and the beginning of customer development. Scrum techniques again include project planning, and reflection and evaluation techniques. Additionally, agile development techniques created around Scrum, Lean Startup and other agile methodologies are important for implementation efforts around technical prototypes and the MVP. As in Sect. 4.3.1 the following descriptions should be considered examples and represent only a couple of the techniques that are helpful during this phase.

The most prominent mid-fidelity prototyping techniques include interactive wireframes and more sophisticated UI prototypes. Interactive wireframes can be used to evaluate interaction and navigation concepts as well as arrangements of content in the software. They can be created with simple sketches on paper adding interactivity by adding content during testing or having movable and interchangeable bits of the prototype. Additionally, using apps like Marvel¹ can create an actual app from sketches that can be send to testers. Digital wireframing tools like Pidoco² or Mockingbird³ provide a variety of building blocks to build slightly more sophisticated wire framed screens and clickable prototypes often allowing a presentation as web-page or smart phone app.

For more sophisticated UI prototypes that provide a sense of the actual app design (e.g., for presentations to management or stakeholders) tools like Keynotopia⁴ enable the team to build clickable UI prototypes in the actual design within Powerpoint or Keynote. Alternatively, HTML Pages can be created for the same purpose. The qualitative and measurement-based testing techniques presented in Sect. 4.2.2 are also usable during this phase.

When the InnoDev Team starts to look for aspects to clarify, to refine the user stories, or to decide on which features to include in the MVP, user story maps are a helpful tool. During user story mapping the functionality and features of the solution concept are transferred to agile user stories, which are then arranged on a User Story Map. Such a map arranges the main activities possible in the software from left to right in an order that makes sense, e.g. in a workflow or by priority (Patton 2009). Additionally, task centric user stories are arranged under the activity they belong to, also arranged from left to right. Tasks that can occur in parallel will be placed vertically under one another. Thus, such maps provide information about the planned functionality of the system under development and its iterations and can be used to identify holes and omissions in a backlog and plan releases that deliver value to user and business.

The Business Model Canvas and the Lean Canvas are valuable tools when it comes to creating, testing and adapting the business model for the software under development. The Business Model Canvas was originally proposed by Osterwalder (Osterwalder and Pigneur 2010) and has since been adapted for various special

¹<https://marvelapp.com/>

²<https://pidoco.com/>

³<https://gomockingbird.com>

⁴<https://keynotopia.com>

cases. One such adaption is the Lean Canvas proposed by Maurya (2012), which is based on the work of Ries (2011).

Technical spikes are a good way to test software libraries and prototype the technology stack. Furthermore, they facilitate the possibility to work out solutions for technical issues or validate the technical feasibility of an idea through a simple implementation that is not aimed to be deliverable.

Once an MVP has usable features, it is a good idea to look for early evangelists (Blank 2007), that is users that are willing to take a risk and use an un-finished product. Such users provide crucial help in product development by their motivation to solve an urgent problem, encouraged by the vision of such a solution in place in the future.

In addition to the Scrum meetings as described in Sect. 4.1, a daily clickthrough of the current prototypes ensures that everyone in the team is up to date on the explored concepts and findings.

4.3.3 Roles

During this phase, the InnoDev Team is responsible for the planning and execution of the development sprints. Ideally the team that was working during the Design Thinking phase will continue during this phase and be extended with additional developers or other team members from areas of expertise as necessary for the software under development (e.g., back end developers, front end developers, database experts, UI developers, UI designers or interaction designers, business experts, or sales and marketing personnel).

(Potential) users and stakeholders will be responsible for testing the different prototypes developed during the Initial Development phase and give feedback on other artefacts and ideas, such as the business model.

Similarly, the project sponsor(s) give(s) feedback on the developed prototypes and the general direction of the project. In addition, they aim to facilitate communication with relevant departments of the their company and advertise the project progress to people interested.

During this phase, the role of the Product Owner (PO) starts to make sense. The PO is the representative of the customer and is responsible for creating the backlog and updating and prioritizing the user stories. For InnoDev, we propose to draw the PO or POs from the team members involved during the Design Thinking phase (e.g., a user researcher or designer trained for this role). A team of POs with a combination of designer, business and developer perspectives can be valuable for larger projects.

The Process Master has the same responsibilities as defined in Sect. 4.2.3.

4.3.4 Deliverables

The main deliverable of this phase is the MVP. It is complemented by other design and technical prototypes that are created for further refinement of the solution as well as a business model. The knowledge gained from developing and testing the

prototypes, the business model and the MVP should lead to further functional, technical and non-functional requirements as well as more refined and new agile user stories.

4.4 Development Phase

The Development phase as illustrated in Fig. 6 is basically an agile development phase making use of the Scrum process framework and Lean Startup validation and scaling techniques. This phase enables the InnoDev Team to work towards a final product and business in incremental steps. Design Thinking is less prominent as in the other two phases. Instead of providing the main activities from its process, it only provides methods from its tool box where applicable.

4.4.1 Activities

The activities during this mode follow a basic software development approach using Scrum project management complemented by Lean Startup validation and scaling. The InnoDev Team focuses on the development of software increments including deployment and maintenance concepts as well as scaling the business. In case a) new features become necessary, b) existing features need to be refined or c) problems arise with existing features, the business model, or the team and their processes Design Thinking activities in the form of smaller workshops or single techniques can be chosen by the team to help them solve the task or problem at hand.

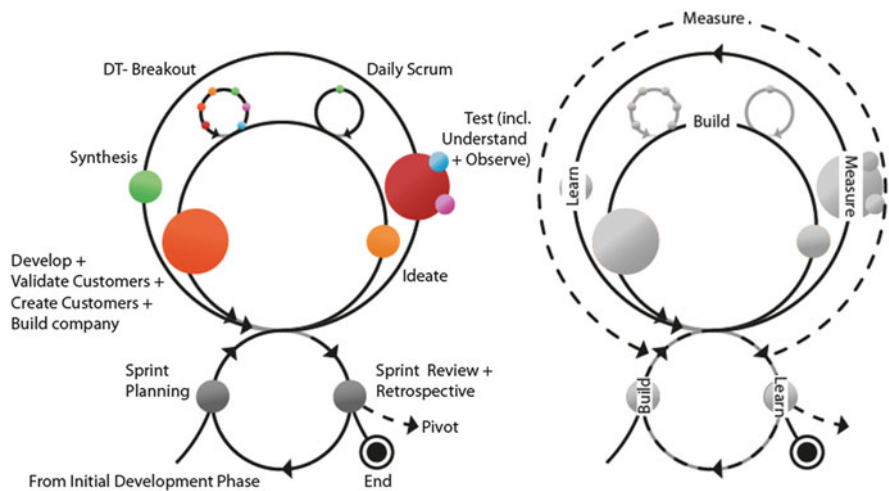


Fig. 6 Overview of the Development phase and representation of the build measure and learn concepts within this phase

4.4.2 Techniques

As this phase aims to further develop the MVP into a fully functioning software product and to develop the business around this product, software engineering techniques and customer development dominate the work. These include practices such as test-driven development, continuous integration, different review techniques to maintain code quality, collective code ownership, and continuous customer testing [compare (Beck 2000; Ries 2011)]. Additionally, techniques from Lean Startup used to scale the business and for validation are helpful. These include making use of actionable metrics, for acquisition, activation, retention, revenue and referral (AARRR), which can be used to assess the product performance and evaluate the product, the business model and the marketing strategy (Maurya 2012). As this phase is still about continuous learning and improvements, a good technique to find out which of two implementations works better A/B or split testing is helpful. It can be used to evaluate different marketing campaigns in different areas or different features or implementations of a feature by splitting the users into groups and providing them with different versions. After acquiring customers, and looking ahead for growth it would be a good move to establish a customer advisory board.

Design Thinking techniques will be used in an ad-hoc manner as necessary and therefore can be drawn from the entire spectrum of techniques, for example the techniques described in Sect. 4.2.2.

4.4.3 Roles

The responsibilities of the InnoDev Team during this mode are similar to those in the preceding mode. They plan and execute the sprints implementing functional software increments. If needed, additional Scrum teams can be added to allow for parallel development.

The (potential) users are again tasked with testing the software increments and giving feedback to changes and feature ideas.

The Project Sponsor is still tasked with facilitating communication with interested departments inside their company and to promote the project progress to interested parties. Additionally, he will give feedback on the developed software increments during reviews.

The Product Owner has similar responsibilities as described in Sect. 4.3.3.

The Process Master has the same responsibilities as during the other modes.

4.4.4 Deliverables

The Development phase focuses on creating tested, working software and developing the corresponding business. Thus, the deliverables are a product that is continuously improving as well as a growing business.

During this phase, all software development should be potentially shippable by the company. This means that the software should adhere to product standards as defined by the company and necessary for the market and the users. Furthermore, it needs to be delivered to the users or customers on a regular basis even if it is not an online product or a mobile app, in which cases a deployment strategy might be necessary.

4.5 Configurations

InnoDev is a software development methodology generally applicable to different company settings such as startups, small and medium-sized enterprises and large organizations. However, differences in a specific context, such as team-size, project goals, product size, level of expertise for the methodology etc. exist and should be targeted by adapting InnoDev to the specific context of a project. In this section, we will describe possible adaptations to the general InnoDev process for specific needs which we believe should be addressed. Our suggestions in this chapter do not present a complete list of possible adaptations but rather stem from our former work and can be extended for other needs and contexts.

4.5.1 Goals

We believe our process to be applicable for the development of innovative and new software products as well as for the incremental and on-going development of existing software products.

When a new product is developed, the challenge used to start into the Design Thinking phase can be formulated accordingly, giving the general area of the product and some context of its users, an example challenge could be: “How might we help elderly people to be more mobile in their everyday life?”. The team should then start to investigate the market for existing products in this area and research user needs by interviewing and observing the user group.

In case existing software will be extended and InnoDev is used to discover new features, the InnoDev Team needs access to the existing software or, ideally, to include members from the development team of the existing software. In such cases, company requirements or product standards may already play a larger role in the beginning of the InnoDev process.

The challenge (or problem statement) should be formulated accordingly, for example: “Discover features for our online-shop that appeal to teenagers!”. The InnoDev Team then starts with the same process but is already more focused on the existing solution and might have access to existing customers for interviews and observations. Furthermore, the development of a business model during the Initial Development phase might not be necessary if the existing business model is sufficient.

4.5.2 Team Size and Setup

The team size has a potential impact on the project. Small teams can be more flexible and more easily able to change their direction, whereas large teams have a greater work force and thus the ability to cover more aspects or produce more results. In the case of working with a bigger team comprised of several smaller teams, it is necessary to scale the InnoDev process, establish a communication structure between teams and some form of a team lead. For ideas on how to scale InnoDev please refer to Sect. 4.6.

The team setup could also be configurable. Ideally, the InnoDev Team stays constant throughout the process to avoid handovers and the not-invented-here syndrome. This state can be achieved by keeping everyone who was involved during the Design Thinking phase in the team for the later phases and only adding personnel as needed, for example if special skills or more development power are necessary. However, that might not always be possible due to people leaving the companies or having other priorities inside the company. Switching teams between phases is therefore possible, however in such a case special care needs to be taken of producing light-weight documentation accompanying each deliverable to make sure the new team understands where ideas are coming from, why features are important, and so on.

4.5.3 Level of Expertise

The InnoDev process can be used by teams that have all the required expertise to use Scrum, Lean Startup, and Design Thinking as well as inexperienced teams who only have some or none of that expertise at their disposal. Experienced teams will be able to choose the right techniques during each activity and decide when to move from one activity to another or when to switch to the next phase by themselves or with the help of the InnoDev Process Master. The decisions to switch between phases can be made in consultation with project leads or managers, should they be established. More inexperienced teams will probably need stronger guidance in making these decisions. In such cases, the Design Thinking phase could prescribe techniques that the team has to run through, as is proposed by de Paula (2015). Another possibility is to structure both the Design Thinking and the Initial Development phase with the help of milestone deliverables as proposed by Vetterli et al. (2013). Figure 7 visualizes a possible distribution of such milestones in the Design Thinking phase and the Initial Development phase. Finally, inexperienced teams can be guided and supported by a team of coaches and experts throughout the process if the necessary budget and manpower are available.

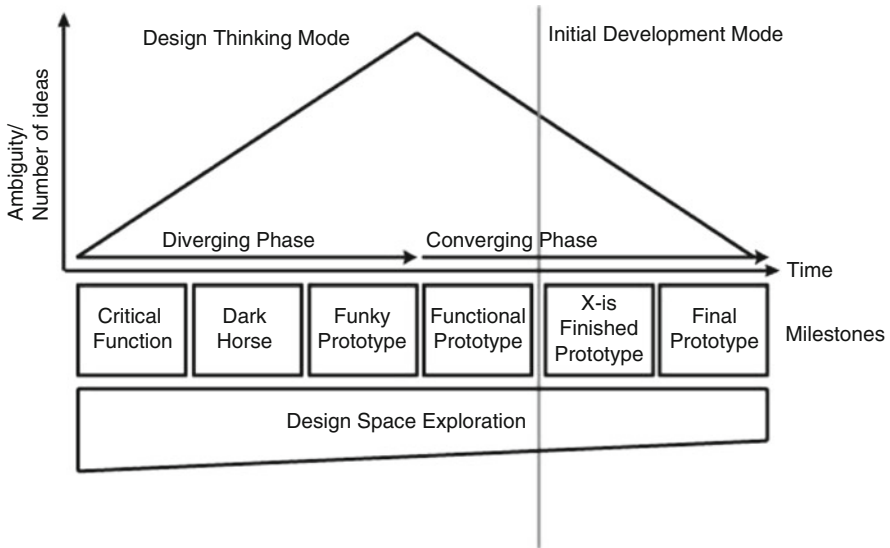


Fig. 7 Milestone concept during the Design Thinking and the Initial Development phase (Häger et al. 2015 adapted from Vetterli et al. 2012)

4.5.4 Company Sizes

The size and structure of a company can have a large impact on a project. While teams in smaller companies can work relatively free and in close cooperation with management. Teams in large organizations might have to report to various managers, fit into the company strategy, adhere to quality and security standards and be subject to audits. In such cases, the Initial Development phase should be used to create specifications of how the product under development will integrate into the company context, including the identification of dependencies with other projects, or possibilities to reuse existing software systems or components in the final implementation. Furthermore, it is possible to set up a transition between the phases of InnoDev in a stage-gate manner, thus allowing for management reporting and approval, as well as audits, before moving to the next phase.

Another smaller aspect that might depend on the company size, is the fit of specific techniques for the teams. For example, the development of a business model is a crucial aspect no matter the company’s size, but different tools are implemented depending on whether a company is large or small. Based on experiences from practice the Lean Startup canvas is useful for startups, while the business model canvas is the right tool for larger organizations.

4.6 *Scaling InnoDev*

If the software under development is large in terms of features and potential user groups (e.g., business software that aims to bundle several business processes for several users into one piece of software), working with one small team might not be sufficient. In such cases, it becomes necessary to scale InnoDev to be able to work with several teams.

Scaling for the Design Thinking phase is scarcely researched so far. One possible way to scale the Design Thinking process has been reported by Häger and Teusner (2014). They describe a multiteam Design Thinking workshop series to kickstart larger software projects. Key elements of their approach are depicted in Fig. 8.

A kickoff workshop introduces the teams to their challenge and if necessary to Design Thinking. During this workshop, all teams fast forward once through the whole Design Thinking process. Following the workshop, the teams have time for teamwork intertwined with further workshops, in which they run through the Design Thinking process again with more time and iterate on their ideas. The workshops allow for an exchange of ideas and feedback as well as communication between the different teams. Furthermore, they provide a possibility to track the progress of the teams. Once the teams have arrived at final ideas and prototypes, Häger and Teusner propose to evaluate ideas and combine them to possible larger pieces of software that can then be further evaluated and developed in follow-up projects. Figure 9 depicts a possible development from ideas to smaller projects and then to a final bigger project and maps those projects to the phases of InnoDev.

Smaller follow-up projects include innovation projects, technical proofs of concept, or developing MVPs for specific user groups and help to clear open technical or conceptual questions and further refine specific ideas. Finally, ideas and outcomes of these smaller follow-up projects will be combined into one bigger software development project.

We believe the approach presented by Häger and Teusner can be integrated into the Design Thinking phase, as the prerequisites, activities, deliverables and roles that are necessary largely overlap. Only the role of the team lead or team leaders would be new and the workshops would form another activity in addition to the activities for each of the participating InnoDev Teams. We believe that the team leaders should be enlisted out of the InnoDev Teams similar to a group of POs in a scaled Scrum environment, thus allowing all teams to be represented. The concept of the follow-up projects also fits well into the Initial Development phase making it possible to scale this phase in the form of multiple projects that refine and test different parts of the final combined products. Ideally the InnoDev Teams from the first phase will be able to continue working on their ideas during the Initial Development phase.

Finally, when moving towards the bigger software development project, teams working in the Development phase can rely on methodologies for Scaling agile software development, such as Large-Scale Scrum (LeSS) (Larman and Vodde 2009, 2010, 2013) or the Agile Scaling Model (Ambler 2009). Both Ambler, and Larman and Vodde (Larman and Vodde 2009, 2010, 2013) present examples and

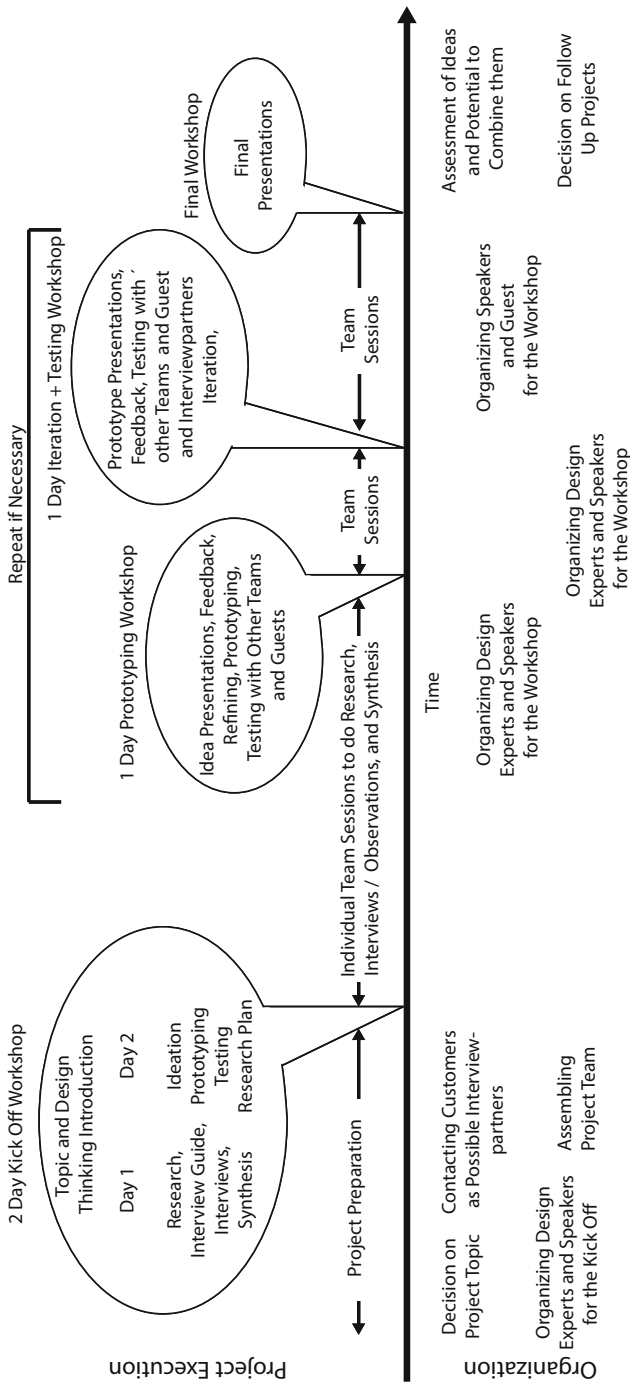


Fig. 8 Scaling Design Thinking for several teams in a series of workshops (Häger and Teusner 2014)

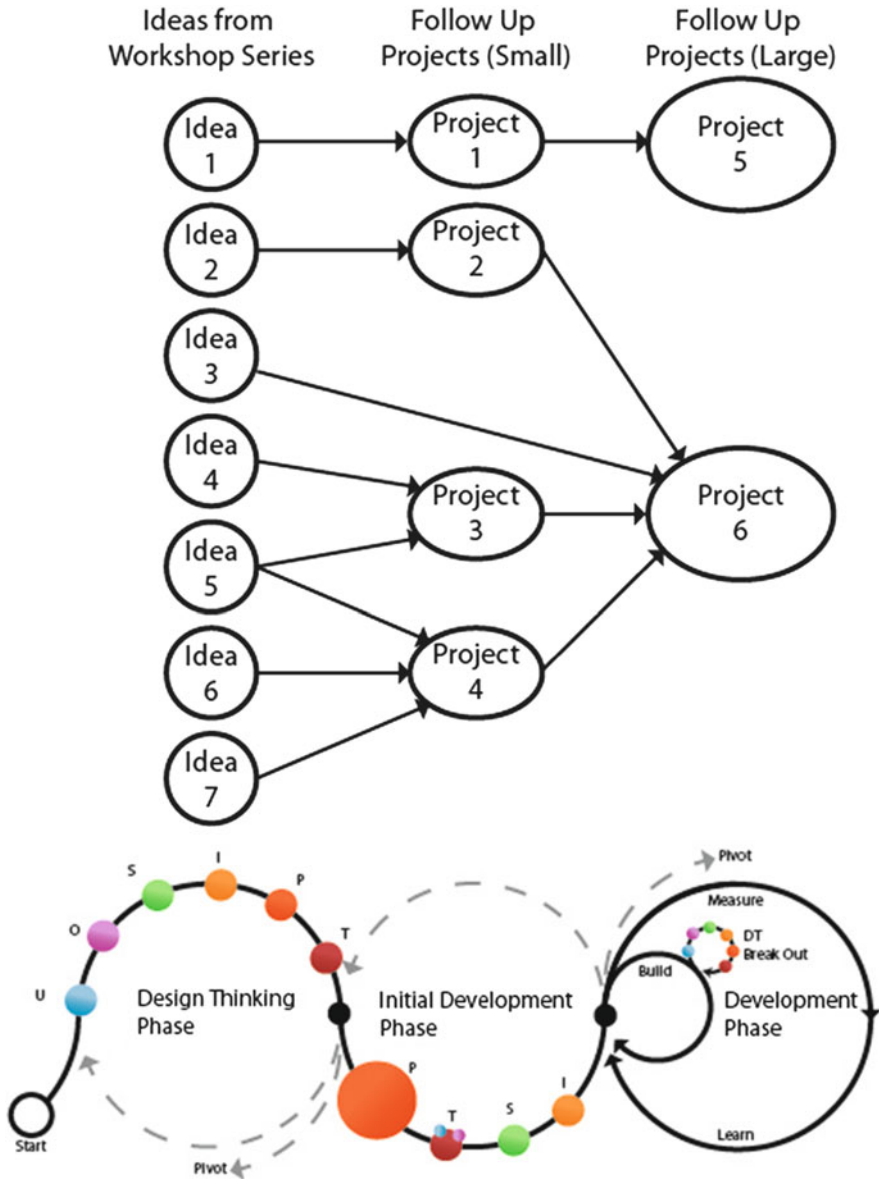


Fig. 9 Development of ideas from the workshop into a series of follow-up projects mapped to the InnoDev phases (adapted from Häger and Teusner 2014)

case studies on how agile processes can be scaled for large project teams and explain appropriate techniques. Example techniques include the Scrum of Scrums or Meta Scrum, communities of practice or learning days.

In the Scrum of Scrum technique, the individual Daily Scrum of all teams is followed by a Daily Scrum of Scrums with an ambassador from each team, who will give a progress report from his team and take back important information to his team members. If necessary, this technique can be used on multiple levels. Communities of practice allow for the exchange of knowledge and ideas or the discussion of problems between people with the same role or type of expertise. In this technique, such groups meet regularly to discuss problems and ideas with each other. Example groups could be design (incl. UI, interaction and visual designers), testing (including various testers), DevOps, or business (incl. management, sales and marketing personnel). Learning days or learning workshops are a way to spread knowledge throughout big teams or companies, in which a team member or a team can share useful techniques, case studies, or other interesting knowledge with other teams.

5 Evaluation

It has been demonstrated that integrating Design Thinking to the software development process enables increased team collaboration (Carlgren et al. 2014), better understanding of the user and product (Liedtka 2015). For this study, we aim to evaluate InnoDev by following a two-phase approach: survey development, and survey application and workshops. Survey Development corresponds to the creation and measurement of an instrument to validate the InnoDev model. Survey application and workshops aim to run workshops that teach InnoDev with different companies and apply the validated instrument to developers, designers and managers from the companies attending the workshop. Additionally the questionnaire will be send out to various companies involved in software development to collect a large sample of responses.

5.1 Survey Development

In order to validate InnoDev, a questionnaire will be developed, validated, tested and applied to developers, designers and managers in software companies. The items in the questionnaire will be developed to answer the following research questions:

- RQ1: To what extend are solutions proposed by InnoDev important to software development companies?
- RQ2: To what extend are solutions proposed by InnoDev already implemented by software development companies?
- RQ3: To what extend can solutions proposed by InnoDev help avoid common flaws in the software development process?

In order to ascertain that our questionnaire is well designed and the items are measuring all aspects of the InnoDev model we will make use of content validity and

face validity checks. A content validity check will be undertaken to ascertain whether the content of the questionnaire is appropriate and relevant to the study purpose. Six experts from the areas of Design Thinking, software development, and survey design will be asked to review the items of our questionnaire to ensure they are consistent with InnoDev. A face validity check verifies whether the questionnaire is appropriate to the purpose of the study and content area. It evaluates the appearance of the questionnaire in terms of feasibility, readability, consistency of style and formatting, and the clarity of the language (Haladyna 2004).

5.2 Survey application and workshops

In order to apply the survey, we will distribute the questionnaire to a wide range of software development companies through appropriate groups on LinkedIn and Xing and through personal contacts. Additionally, we will design a workshop aimed at walking companies through each step of the InnoDev model in order to teach the InnoDev process and to apply our questionnaire to the attendees. The workshop will be designed to accommodate the needs of a wide variety of software company roles. For example, designers, developers, managers and stakeholders who have a keen interest in software solution innovation. Participants in the workshop will benefit from the applied learning of new techniques through the InnoDev model. The workshop will encourage participants to identify (a) opportunities for new software solutions in their companies and / or (b) enhance existing software and service offerings by aligning solutions to specific customer needs. Specifically the workshop will demonstrate how InnoDev can support companies to align Design Thinking, product development, customer development and value realisation for new software products and services.

Before running the workshop we will conduct a pilot test of the workshop. This pilot test will be a condensed version of the final workshop. The participants of the pilot workshop will be asked to answer a questionnaire after the pilot workshop to evaluate its quality. The questionnaire will include a simplified set of criteria for evaluation as used by (Pigosso et al. 2013), which are: utility, consistency, simplicity, clarity, coherence, instrumentability and forecast. Based on the results of the pilot workshop the concept will be refined. We plan to run at least two workshops with startup companies from Galway, Ireland and Berlin, Germany.

6 Outlook and Summary

The aim of this chapter has been to describe in detail a framework that combines Design Thinking, Lean Startup and Scrum for software development that can deliver the innovative customer-oriented products and services required by competitive companies. InnoDev was developed based on existing models from the literature

that have been recently studied. The findings from our study are a step towards aligning relevant research in order to enable the next generation of research on the software development process.

First, we describe the three phases of InnoDev and how Design Thinking, Lean Startup and Scrum interact with each other: The Design Thinking phase, the Initial Development phase and the Development phase. All three phases are in line with what other researchers (Grossman-Kahn and Rosensweig 2012; Hildenbrand and Meyer 2012) claimed to be relevant to a software development process. The three phases essentially center on trying and learning each using different tools and techniques. Design Thinking aims to understand and learn about problems and user needs in order to derive solutions and product options. Lean Startup aims to learn about business strategies and scaling options and Scrum about development options and further directions during software development with changing requirements.

Further, we propose that Scrum is used as the overall project management method underlying all phases of InnoDev. In particular, we propose companies use Scrum to structure the Design Thinking phase in order to let teams get a feeling for the duration and value of Design Thinking activities, and to enable them to better structure their creative work.

Our findings provide complementary perspectives regarding software development strategies, roles and techniques. Future work could expand our findings and evaluate InnoDev in an industry scenario, which might help us better understand how to enhance the synergy between the approaches.

This study advances knowledge of Design Thinking and software development by providing a detailed description of a tool that combines best practices for creating more innovative software products. The results of this investigation can help managers to evaluate their software development process and thereby improve its effectiveness and create more efficient user-driven solutions.

References

- Ambler, S. W. (2009). The agile scaling model (ASM): Adapting agile methods for complex environments. *Environments*.
- Beck, K. (2000). *Extreme programming explained: Embrace change*. Boston: Addison-Wesley Professional.
- Beverland, M., & Farrelly, F. J. (2007). What does it mean to be design-led? *Design Management Review*, 18(4), 10–17.
- Blank, S. G. (2007). *The four steps to the epiphany: Successful strategies for products that win*. California: S. G. Blank.
- Carlgrén, L., et al. (2014). Exploring the use of design thinking in large organizations: Towards a research agenda. *Swedish Design Research Journal*, 1(14), 47–56.
- Chin, K.-S., et al. (2008). Group-based ER–AHP system for product project screening. *Expert Systems with Applications*, 35(4), 1909–1929.
- d.school Stanford. Bootcamp Bootleg.

- de Paula, D. F. O. (2015). Model for the Innovation Teaching (MoIT): um modelo baseado em Design Thinking, Lean Startup e Ágil para estudantes de graduação em computação.
- de Paula, D. F. O., & Araújo, C. C. (2016). *Pet empires: Combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment*. In International Conference on Human-Computer Interaction (pp. 30–34). Springer.
- de Paula, D. F. O., et al. (2014). *Building a quality mobile application: A user-centered study focusing on design thinking, user experience and usability*. In: International Conference of Design, User Experience, and Usability (pp. 313–322). Springer.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *The Scrum Primer: A lightweight guide to the theory and practice of scrum*. Version 2.0.
- Derby, E., et al. (2006). *Agile retrospectives: Making good teams great*.
- Dingsøyr, T., et al. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221.
- Dobrigkeit, F., & de Paula, D. F. O. (2017). *The best of three worlds – The creation of InnoDev a software development approach that integrates design thinking, scrum and lean startup*. In Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 8, Human Behaviour in Design (pp. 319–328). Vancouver, Canada.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859.
- Eisenmann, T. R., et al. (2012). Hypothesis-driven entrepreneurship: The lean startup. *Harvard Business School Background*. Note 812-095.
- Erickson, J., et al. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4), 88.
- Freeman, R. E. (2010). *Strategic management: A stakeholder approach*. Cambridge: Cambridge University Press.
- Frishammar, J., et al. (2011). Beyond managing uncertainty: Insights from studying equivocality in the fuzzy front end of product and process innovation projects. *IEEE Transactions on Engineering Management*, 58(3), 551–563.
- Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods Renaissance Software Consulting*, 3, 1–3.
- Grossman-Kahn, B., Rosensweig, R. (2012). Skip the silver bullet: Driving innovation through small bets and diverse practices. *Leading Through Design*, 815–829.
- Gutzwiller, T. A. (2013). *Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen*. Berlin: Springer.
- Häger, F., & Teusner, R. (2014). *Scaling design thinking – Using a multi-team design thinking workshop to kick start software projects*. Presented at the Design Thinking in Business Information Systems Workshop at the European Conference on Information Systems, Tel Aviv.
- Häger, F., et al. (2015). DT@Scrum: Integrating design thinking with software development processes. In H. Plattner et al. (Eds.), *Design thinking research* (pp. 263–289). Heidelberg: Springer.
- Haladyna, T. M. (2004). *Developing and validating multiple-choice test items*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Hildenbrand, T., & Meyer, J. (2012). Intertwining lean and design thinking: Software product development from empathy to shipment. In A. Maedche et al. (Eds.), *Software for people: Fundamentals, trends and best practices* (pp. 217–237). Heidelberg: Springer.
- Hirschfeld, R., et al. (2011). Agile software development in virtual collaboration environments. In *Design thinking* (pp. 197–218). Heidelberg: Springer.
- Kerth, N. L. (2000). *Project retrospectives – A handbook for team reviews*. New York: Dorset House.
- Komus, A. (2017). Status quo agile. Stud. Zur Verbreit. Nutzen Agil. Methoden Hochsch. Kobl.
- Kua, P. (2013). *The retrospective handbook*. E-book Available <https://leanpub.com/retrospective-handbook>.
- Larman, C., & Vodde, B. (2009). *Scaling lean & agile development*.

- Larman, C., & Vodde, B. (2010). *Practices for scaling lean & agile development: Large, multisite, and offshore product development with large-scale Scrum*. Upper Saddle River, NJ: Addison-Wesley.
- Larman, C., & Vodde, B. (2013). Scaling agile development. *CrossTalk*, 9, 8–12.
- Liedtka, J. (2015). Perspective: Linking design thinking with innovation outcomes through cognitive bias reduction. *Journal of Product Innovation Management*, 32(6), 925–938.
- Lindberg, T., et al. (2011). Design thinking: A fruitful concept for IT development? In C. Meinel et al. (Eds.), *Design thinking* (pp. 3–18). Heidelberg: Springer.
- Maurya, A. (2012). *Running lean: Iterate from plan A to a plan that works*. Sebastopol: O'Reilly Media.
- Müller, R. M., & Thoring, K. (2012). *Design thinking vs. Lean startup: A comparison of two user-driven innovation strategies*. In *Leading innovation through design proceedings of the DMI 2012 International Research Conference 2012*, 8–9 August, Boston. Boston: Design Management Institute.
- Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*. Hoboken: Wiley.
- Patton, J. (2009). Telling better user stories – Mapping the path to success. *Better Software*, 11(7), 24.
- Pigosso, D. C. A., et al. (2013). Ecodesign maturity model: A management framework to support eco-design implementation into manufacturing companies. *Journal of Cleaner Production*, 59, 160–173.
- Ries, E. (2011). *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Business.
- Schwaber, K. (1997). Scrum development process. In *Business object design and implementation* (pp. 117–134). London: Springer.
- Schwaber, K., & Beedle, M. (2001). *Agile software development with scrum*.
- Schwaber, K., & Sutherland, J. (2013). *The scrum guide – The definitive guide to scrum: The rules of the game*.
- Scrum Alliance. (2016). *The 2016 state of scrum report*. Scrum Alliance.
- Thoring, K., & Müller, R. M. (2011). *Understanding design thinking: A process model based on method engineering*. In *DS 69: Proceedings of E&PDE 2011, the 13th International Conference on Engineering and Product Design Education*, London, UK, 08.-09.09.2011.
- Verganti, R. (1997). Leveraging on systemic learning to manage the early phases of product innovation projects. *RD Management*, 27(4), 377–392.
- Version One. (2017). 11th annual state of agile report. Technical report, Version One.
- Vetterli, C., et al. (2012). Initialzündung durch Embedded Design Thinking – Ein Fallbeispiel aus der Finanzindustrie und wie dadurch ein Wandel in der Innovationskultur einer IT-Abteilung eingeleitet wurde. *OrganisationsEntwicklung: Zeitschrift für Unternehmensentwicklung und Change Management*, 31(2), 22–31.
- Vetterli, C., et al. (2013). *Jumpstarting scrum with design thinking*.
- Vilkkki, K. (2010). When agile is not enough. In *Lean enterprise software and systems* (pp. 44–47). Heidelberg: Springer.
- Wölbling, A., et al. (2012). Design thinking: An innovative concept for developing user-centered software. In *Software for people: Fundamentals, trends and best practices* (pp. 121–136). Berlin: Springer.