Cherry Picking - Agile Software Development Teams Applying Design Thinking Tools

Franziska Dobrigkeit[(✉)], Christoph Matthies, Philipp Pajak, and Ralf Teusner

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
`franziska.dobrigkeit@hpi.de`

Abstract. Design Thinking (DT) is an established approach to conceptualize software products before starting the product development work. Research suggests that software development can benefit from a continuous integration of DT throughout Agile development processes. However, practitioners and researchers lack an in-depth understanding of which tools from the ever-growing DT toolbox are suited to support software development teams and their processes and how these tools can be applied to the teams' daily work. As initial steps towards closing this knowledge gap, we present our experiences from testing five different DT tools from a previously developed toolbox with four Agile software development teams. Each team chose three tools to apply to their product, problem, and context during a workshop. We present summarised findings regarding the use cases, benefits, and challenges of these tools as experienced by the participants. Overall, the teams welcomed the DT tools and were able to independently apply them to achieve the desired effects, e.g., to highlight user needs, find product issues, and discover team challenges.

Keywords: Design Thinking · Agile · Software development teams

1 Purpose

Research on Design Thinking (DT) indicates that it can positively impact software development activities, e.g., by facilitating a deeper understanding of the product, the users, and their needs [1,2,15,16], and increasing team collaboration [2,4]. Accordingly, various researchers propose integrating aspects of DT with modern Agile software development [9,11,17]. While the research is often focused on DT as an approach to support initial software design phases, some researchers suggest continuous use through all stages of software development [7,17]. Within such approaches, DT tools benefit Agile teams in later process stages, aiding during their ongoing development activities [5]. However, there is still a lack of understanding of which DT tools can support agile development teams. Accordingly, our research focuses on empowering Agile team members and DT novices with suitable DT tools and evaluating their experiences and collected perceptions (cmp. [5,6]). Towards this goal, we developed

the *DT@IT Toolbox* in previous research [8,13]: a collection of 12 DT tools targeted at supporting software development teams in their work (freely available for download at https://hpi-epic.github.io/dt-at-it-toolbox/).

2   Research Method

Within this study, we employ a convergent mixed method research design [3] to collect empirical evidence for a total of 5 of the tools from the DT@IT Toolbox. We evaluate the tools with four teams from German Small and Medium Enterprises (SMEs). We selected the teams based on company size, self-reported use of an Agile development process, low self-reported knowledge of DT, and colocation to allow for an in-person workshop at the company site. We met with each teams' lead to introduce our toolbox and research and asked them to select three of the twelve tools for their team based on what made the most intuitive sense for their current use cases and problems. Next, we met at the team's site and conducted a three-and-a-half-hour workshop, in which the teams applied each of the three chosen DT tools to issues or topics relevant to their project. We introduce the tools with the help of the worksheets from the DT@IT Toolbox. Each worksheet includes information on required prerequisites, the time needed, an explanation of the tool, and examples from a software development context. The workshops were structured as follows:

1. 15 min introduction to Design Thinking and our research
2. 60 min Application of first DT Tool
   (a) Handing out the worksheet from our toolbox
   (b) Answering comprehension questions
   (c) Application of the DT tool to a problem or topic of the teams choice
   (d) Filling out a tool evaluation survey
3. 60 min Application of second DT tool (detailed steps see 2. a–d)
4. 60 min Application of third DT tool (detailed steps see 2. a–d)
5. 15 min Group interview.

   During the workshops, we took part as participant observers and employed four forms of data collection to capture the participants' experience with the DT tools: The facilitating researcher took field notes on the teams' process including, use-case chosen, discussions in the team, results, challenges. Additionally, the researcher took pictures of the resulting artifacts, e.g., sketches or whiteboards. At three points of the workshop, we asked participants to fill out a survey to evaluate the DT tool they had just experienced. Items in the survey asked for the general experience with the tool, what they liked and disliked, further possible and impossible use-cases, and feedback on the worksheet. At the end of each workshop, we conducted a semi-structured group interview asking them to provide feedback on the workshop, the worksheets, and the tools. Thus, we carried out four semi-structured group interviews lasting between 15 to 20 min. The facilitating researcher took notes during each interview. After conducting all workshops, we analyzed our notes from observations and interviews and the survey answers. We iteratively coded and clustered the observation notes, the

interview notes, and the survey responses to derive connections, patterns, and juxtapositions [10]. Thus we derived the perceived benefits, perceived challenges, and use cases presented in Sect. 3.

3   Findings

We evaluated five of the twelve toolbox tools, namely A Beginners Mind, the Customer Journey Map, 30 s Sketch, 5 Whys, and Five Finger Feedback. Table 1 depicts the use cases, benefits, and challenges we identified for each tool from our empirical data. Overall the participating teams welcomed the additional tools and the different working styles. The toolbox worksheets enabled the participating teams to independently apply the tools and achieve the desired outcomes, e.g., discovering product issues, generating and discussing ideas, or learning something about their product, users, and team members. These findings are in line with similar research on UX methods [14] and suggested benefits of DT [1, 2, 4, 12, 15, 16].

Table 1. Summarized findings for each DT tool evaluated as part of this study

Use cases	Benefits	Challenges
A beginner's mind		
• feature design • UI design • framework choice • architecture choice • algorithm understanding	• explaining things simply • better understanding of problem or topic • abstraction of problem & technical knowledge	• too abstract for some • a beginner is required
Customer journey map		
• find & tackle UX issues • document features • document acceptance tests	• easy to take the user's perspective • helps uncover flaws in UI and process flow	• personas are required • finding the right scope • required time • keeping artifacts updated
30 second sketch		
• GUI-focused problems • searching for different perspectives	• easy to do • easy to dispose • fast results	• low confidence in drawing skills might be a barrier
Five whys		
• finding core problem during bug fixing	• allows to dig deeper than intuitive first answer • sensitizes for simplified answers & root causes	• might feel invasive to the questioned person
Five finger feedback		

• personal feedback
  • with managers
  • on a task or project
• group feedback
  • in retrospectives
  • after longer meetings
	• easy to do
	• easy to remember
	• feedback on different aspects
		• accepting feedback
		• distinguishing feedback categories can be hard
		• difficult with broad topic
		• not suitable for large groups

4   Research and Practice Implications

The results from this study support our hypothesis that DT tools can benefit Agile teams during later stages of their development process, as they are aiding in their ongoing development activities and challenges. However, we were only able to evaluate five selected DT tools with a limited number of teams. Accordingly, further research with additional tools and teams is necessary to provide more evidence. Additionally, we only evaluate the direct effects of DT tool usage, and more research is required to collect and assess long-term results.

For Agile practitioners, we provide a low-friction possibility to get acquainted with suitable tools with our DT toolbox and the accompanying materials. The use cases and benefits identified within this research can guide the selection of fitting tools for a given situation, and the identified challenges provide hints of what to prepare beforehand or what to be aware of during the application of a specific tool. Our toolbox can serve as a stand-alone support for software development teams. In this case, teams that want to further their DT knowledge beyond the application of single tools will require additional in-depth explanations on DT and the associated concepts and background. Alternatively, our toolbox can support more holistic integration approaches (cmp. [9,11,17]) by providing DT use cases and tools to apply in later development stages.

5   Contributions

We contribute empirical evidence on how and when a selection of five DT tools can support Agile software development teams in their daily work. We summarize the employed use cases and the benefits and challenges experienced by the groups for each tool. The participating teams independently and successfully applied the introduced DT tools with the help of worksheets from the DT@IT Toolbox [8]. Thus, our findings suggest that DT novices can benefit from applying single DT Tools without extensive introduction or training in DT concepts. Furthermore, our results indicate that one-time use of the DT tools already provided the reported benefits. However, these are most likely not permanent without repeating similar activities. Additionally, the use of single DT tools only provides a limited perspective on DT as it does not support developing a DT mindset or an understanding of the DT process. Nonetheless, we conclude that the evaluated worksheets from the DT@IT toolbox and the identified use cases provide a low-barrier entry-point to DT, allowing agile development teams to gain first-hand experiences with DT tools and experiencing their benefits.

References

1. Brown, T.: Design thinking. Harvard Bus. Rev. **86**(6), 84 (2008)
2. Clark, K., Smith, R.: Unleashing the power of design thinking. Des. Manag. Rev. **19**(3), 8–15 (2008)
3. Creswell, J.W., Clark, V.L.P.: Designing and Conducting Mixed Methods Research. Sage Publications, Thousand Oaks (2017)
4. De Paula, D., Dobrigkeit, F., Cormican, K.: Design thinking capability model (DTCM): a framework to map out design thinking capacity in business organisations. In: 15th International Design Conference, pp. 557–566 (2018)
5. Dobrigkeit, F., de Paula, D.: Design thinking in practice: understanding manifestations of design thinking in software engineering. In: Proceedings of the 27th ACM Joint Meeting-ESEC/FSE 2019, pp. 1059–1069 (2019)
6. Dobrigkeit, F., de Paula, D., Carroll, N.: InnoDev workshop: a one day introduction to combining design thinking, lean startup and agile software development. In: 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), pp. 1–10. IEEE (2020)
7. Dobrigkeit, F., de Paula, D., Uflacker, M.: InnoDev: a software development methodology integrating design thinking, scrum and lean startup. In: Meinel, C., Leifer, L. (eds.) Design Thinking Research. Understanding Innovation, pp. 199–227. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-97082-0_11
8. Dobrigkeit, F., Pajak, P., de Paula, D., Uflacker, M.: DT@IT toolbox: design thinking tools to support everyday software development. In: Meinel, C., Leifer, L. (eds.) Design Thinking Research. Understanding Innovation, pp. 201–227. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-28960-7_13
9. Grossman-Kahn, B., Rosensweig, R.: Skip the silver bullet: driving innovation through small bets and diverse practices. Lead. Des. **18**, 815–829 (2012)
10. Harding, J.: Qualitative Data Analysis from Start to Finish. Sage, Thousand Oaks (2013)
11. Hildenbrand, T., Meyer, J.: Intertwining lean and design thinking: software product development from empathy to shipment. In: Maedche, A., Botzenhardt, A., Neer, L. (eds.) Software for People. Management for Professionals, pp. 217–237. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31371-4_13
12. Holloway, M.: How tangible is your strategy? How design thinking can turn your strategy into reality. J. Bus. Strategy **30**(2/3), 50–56 (2009)
13. Pajak, P., Dobrigkeit, F.: DT@IT toolbox: initial release. Version v1.0 (2021). https://doi.org/10.5281/zenodo.4602920
14. Pedersen, T.Ø.: UX toolbox for software developers. Ph.D. thesis (2016)
15. Porcini, M.: Your new design process is not enough-hire design thinkers! Des. Manag. Rev. **20**(3), 6–18 (2009)
16. Ward, A., Runcie, E., Morris, L.: Embedding innovation: design thinking for small enterprises. J. Bus. Strategy **30**(2), 78–84 (2009)
17. Ximenes, B.H., Alves, I.N., Araújo, C.C.: Software project management combining agile, lean startup and design thinking. In: Marcus, A. (ed.) Design, User Experience, and Usability: Design Discourse. LNCS, vol. 9186, pp. 356–367. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20886-2_34