The Use of Design Thinking for Requirements Engineering

An Ongoing Case Study in the Field of Innovative Software-Intensive Systems

Jennifer Hehn
Institute of Information Management
University of St.Gallen
St.Gallen, Switzerland
jennifer.hehn@unisg.ch

Falk Uebernickel
Institute of Information Management
University of St.Gallen
St.Gallen, Switzerland
falk.uebernickel@unisg.ch

***Abstract*—Requirements Engineering (RE) has been aiding software-intensive development projects for quite some time now. However, today's projects growingly demand for agile and human-centered approaches to discover and meet the often fuzzy needs of the various stakeholders involved. In that regard, Design Thinking (DT) has become one of the most promising methods to address wicked problems and define innovative solutions. We see potential to combine both, the strongly human-oriented working mode of DT with the more formal, technology-driven world of RE, to develop human-centered solutions more effectively. Yet, little is known how such an integration could be realized and which concrete benefits and challenges to expect. To better understand this endeavor, we conduct a longitudinal case study to identify how DT and RE can work together in an agile development setting from the very beginning of idea conceptualization to market-ready implementation. In this paper, we summarize our research protocol and present first findings how Design Thinking can complement current RE practices through meeting known challenges encountered by the RE community of practitioners. We provide a better understanding of the multi-faceted potential of DT for RE for both, scholars and practitioners, and describe open issues and planned future steps in our study.**

***Index Terms*—design thinking, requirements engineering, agile development.**

I. Introduction

Requirements Engineering (RE) has been aiding software-intensive development projects for quite some time now. Today's projects growingly demand for agile approaches to manage complex challenges in rapidly changing environments and for human-centered ways to discover and meet the often fuzzy needs of the various stakeholders involved [1]. Agile software development methods, like Scrum, have gained widespread popularity and have significantly contributed to the way development projects are conducted, especially in changing environments [2]. However, when the real-world problem is "wicked" [3], Design Thinking (DT) has gained recognition as a structured problem-solving approach that builds upon interdisciplinary team work, the exploration of human needs, rapid prototyping, and iterative learning cycles in the early stages of product, service, and system development processes [4], [5]. This rather diverging approach to problem-solving is significantly different from the more converging approaches of RE practices in most software-intensive projects [6]. To enhance innovation efforts, large corporations like IBM [7], Deutsche Bank [8], and SAP [9] have integrated DT principles into their corporate DNA with proven success.

Existing academic studies encourage these efforts and acknowledge DT not only as valuable support for RE practices, but also as a "modern form of requirements engineering" [10:288]. As one of the first, Vetterli et al. [11] pointed out that DT in combination with RE could lead to new ways of working and bring together both approaches. Researchers with a content-view ("what" value-added does DT provide) examine DT for RE mostly in specific domains like learning environments [12], health care [6], or social innovation [13] and recognize its value in terms of product quality, process speed, and user acceptance. Researchers with a more process-oriented view ("how" does DT add value), investigate the integration of DT with existing software development techniques and prototype-based programming. Häger et al. [14], for example, conducted experiments with three operation modes of 'DT@Scrum', providing a different ratio of DT and Scrum activities, and found evidence for higher desirability and innovation potential of the solution that was developed in an integrated mode.

We see potential to combine both, the strongly human-oriented working mode of DT with the more formal, technology-driven world of RE, to develop innovative solutions more effectively. Despite the practical and theoretical advances to confirm the potential benefit of DT for RE, little is yet known about how such an integration could be realized and which benefits and challenges to expect. Specifically, knowledge on the specific role of DT for RE practices still remains unclear. In this context, we are conducting a study to understand how DT is used for RE in a project setup. Accordingly, we aim to answer the following research questions:

RQ1: *How is DT used for RE in innovative software-intensive development projects?* The first RQ examines the actual use of Design Thinking for RE from the very beginning of problem definition to market-ready implementation in an agile development setting. By this, we intend to generate a comprehensive foundation to answer RQ2: *What can be learned from DT for RE practices?* This question examines the learning

2332-6441/18/$31.00 ©2018 IEEE
DOI 10.1109/RE.2018.00-18

potential of DT for RE to generate findings, how DT can complement current RE practices addressing known challenges encountered by the RE community of practitioners.

To investigate our research questions, we set up a longitudinal case study. We use an exploratory approach by analyzing the use of DT for RE against the background of existing scientific findings of DT and challenges in RE. We firmly believe, that a more profound understanding of DT would enable both communities, RE and DT, to better evaluate its application purpose and potential for eliciting and specifying requirements and guide companies in a more meaningful way for integrating both disciplines [15].

In this paper, we summarize our research protocol and present preliminary results, open issues, and planned next steps. In Section II, we briefly summarize background information on DT and RE. Section III elaborates on the research methodology. In Section IV we disclose preliminary results while Section V provides first conclusions and an outlook for future work.

II. Research Background

To the extent necessary for setting the background of the ongoing study presented in this paper, we provide background information about the general concept of DT. In addition, we summarize the challenges of agile RE practices, as we leveraged them to build a research framework to analyze RQ2.

A. DT as a Problem-Solving Approach

The origins of DT root back to the late 1960s when design academics like Simon [16] investigated cognitive models as the basis for design activities and transformed them into normative guidelines for creative problem-solving. These studies have broadened the scope of design from the aesthetics of a product towards a way of thinking and doing that can be universalized for various disciplines. One popular re-definition has managed to generate substantial hype in business press by assigning design (thinking) a new role in innovation practice. In this context, DT is defined as a human-centered approach to innovation that draws from the designer's toolkit to integrate the needs of people (desirability), the possibilities of technology (feasibility), and the requirements for business success (viability) [4]. The dimension of desirability (what people want and need) anchors in a deep empathy for users and is applied by involving relevant stakeholders systematically throughout the entire development process, both via direct dialog and non-obtrusive observation methods. Diverse design techniques help facilitating the creative transformation of user knowledge and insights into new concepts. Subsequently, feasibility and viability are integrated and explored. The lens of feasibility (how technology can help), therefore, demands an exploration of organizational capabilities and technological options to translate the human-centered requirement into actual products and services. Assessing the third dimension of business viability (what is financially sustainable) entails evaluating market opportunities and their compliance with the business objectives of the organization.

In the context of product and software development, three perspectives on DT have emerged [17]: (1) DT as an iterative sequence of process steps (e.g., empathize, define, ideate, prototype, test [18]), (2) DT as a toolbox with a loose bundle of methods for adaptive support (e.g., persona, interviews, feedback capture grids [18]), and (3) DT as a mindset with human-centered principles that should be internalized (e.g., focus on human values, bias toward action [17]). All three perspectives have strong overlaps, however, result in different conceptualizations on a practical level. In terms of combining DT with RE, this differentiation is not just a semantic concern, but can strongly influence its integration and even determine its overall success. We will address these dimensions in the results section, mainly when answering RQ1.

B. Challenges of Agile RE Practices

Academics have generated comprehensive lists of challenges in RE in agile setups. We present a summary of themes in TABLE I.

TABLE I. Challenges of Agile RE Practices

	Challenge Themes	Implications for RE
C1	Minimalistic documentation	Requirements in minimalistic documentations are difficult to trace back [2], [19]
C2	Problems with customers or users	Hindering access to and communication with customers slows down the process of clarifying requirements [2], [19], [20]
C3	Neglect of non-functional requirements	User stories mostly focus on system and product features, not non-functional requirements (e.g. security, usability) [19]
C4	Inappropriate architecture	Due to short planning time horizon, the architecture might be inadequate or inappropriate [19]
C5	Tacit requirements knowledge	Most knowledge often stays tacit as agile practices rely on highly skilled people [2], [20]
C6	Imprecise effort estimates	Estimates of time and cost are difficult due to the agile project character [2], [19]
C7	Difficulties in the prioritization of requirements	Focus on immediate business value as prioritization focus might cause neglect of system related requirements [2], [19]

III. Research Methodology

Our aim is to investigate how DT is used for RE in software-intensive development projects and what can be learned accordingly. We adopt a qualitative approach since we want to gain an in-depth understanding of DT for RE as a socio-technological activity [21]. As grounding methodology, we believe that an exploratory case study [22] is best suited to elaborate specific challenges encountered by the RE community of practitioners to theorize potential concepts in the future. Thus, we follow a longitudinal case study approach to gain richer and more contextualized information of the usage of DT for RE over time in a real-life project setting [21], [22]. In the next subsections, we describe our case selection, data collection, and data analysis.

A. Case Selection

To identify an appropriate case, we got access from a Swiss-German consultancy to their project portfolio over the last 5

years. We evaluated their projects according to the fit to our research objective and questions. Specifically, we were looking for (1) the application of DT for RE practices, (2) the development of a new and innovative software system, and (3) an end-to-end setting, from the beginning of conceptualization until market-ready implementation. In 2016, we identified the project case at hand as particularly suitable. The project is based in the utility industry in Europe. The energy sector is currently under heavy pressure (in Europe) because of the need to transform from nuclear and coal power generation towards new and more sustainable ways of generating power like solar power (photovoltaic), water, or wind energy. In addition, this industry is attacked by digital innovations and dealing with changing customer behavior. Smart home technologies like Nest or electric vehicles like Tesla demand for digital solutions and innovation from utilities. TABLE II provides an overview of the characteristics of the case company which is one of the leading utility companies in the world that has to develop a digital transformation strategy.

TABLE II. OVERVIEW OF CASE COMPANY

Sector / Industry	Utility
Employees	> 40.000
Turnover (Billion Euro, year 2015)	> 100
Type of products	- Renewable Energy - Installations (Heat, Solar, etc.) - Energy networks - Digital platforms

In this context, project *Falcon* (anonymized) was launched in 2016 to explore and exploit possibilities of digital platforms in the energy sector, without knowing what the final system would be or look like. Thus, from the beginning on, the project was set up to be highly agile, innovative, and human-centered. On the backdrop of these pre-requirements, the management team decided to apply DT and Scrum as leading methodologies.

B. Data Collection

Since the beginning of the project in August 2016, the data were collected via multiple sources of evidence for the purpose of data triangulation, construct validity, and reliability [21]. We compiled (1) data from contextual interviews, (2) participant-observations, (3) project documentations and presentations, and (4) physical artifacts. First, interviews were conducted with all project team members to gain a comprehensive understanding of each project phase and from all perspectives. Based on the aforementioned criteria and the relevance to our research question, the interview focus was the usage of DT as a mindset, process, and toolbox and its impact on RE. We mainly asked questions about the way of eliciting and capturing needs and requirements, investigating the methods as well as perceived challenges and benefits of DT for RE. Since we are still interviewing project members (as the project team is growing), we leverage information from interviews with the project lead (1), the user research and design team (4), the business model team (2), and the scrum master and software engineering team (3) for the preliminary results of our study. All interviews lasted between 90 and 120 minutes. For confidentiality reasons, four out of our ten interviews were not audio taped, while the rest was. Second, as our goal is to capture the actual RE activities in the project, beyond self-reported practices or official processes, we not only conducted every interview in situ, but also leverage participant-observations in form of meeting protocols and memos for a more objective picture. We expected to gain mainly process-oriented insights about applying DT for RE. The researchers were on site, on average, every second or third week to interact with the project team on a continuous basis. Third, project documentations mainly included pictures and presentations about intermediate results and project progress as well as the access to the project's Jira and Confluence system (currently around 1200 user stories and 7 epics). Fourth, we gathered all physical and digital artefacts, i.e., prototypes ranging from low, non-technical to high resolution prototypes (13). In DT, elicited requirements are expressed via (throwaway) prototypes, which appoints them a vital role in our research evaluation.

C. Data Analysis

We relied on a team-based research approach and applied thematic coding as the study's analysis strategy [21]. Each document was independently analyzed by two researchers to avoid subjective interpretation and enhance the validity of our study [22]. We applied descriptive, in vivo, as well as process coding to cover timing aspects of the material [21]. We examined the raw data (interview transcripts, documentations, participant-observations, and artifacts) in reference to the dimensions of TAB I, yet, always keeping an open mind to extend the lens of these challenges [22]. We constantly cross-examined the constructs in different interviews. Thus, we were able to understand multiple viewpoints as well as reconcile and integrate them. To demonstrate rigor, two researchers discussed emerging patterns and revised them to themes, together in an iterative approach. Furthermore, we were in constant contact with the project team to challenge our emerging concepts.

IV. PRELIMINARY RESULTS

This section presents preliminary findings for both research questions, with a focus on elaborating findings for RQ1.

A. How is Design Thinking used for RE in innovative software-intensive development projects?

This project was initiated based on an idea for a platform around the topic Photovoltaic and Battery (PV&B) for private homes. As the original problem statement was rather vague, the project management decided to apply DT to better understand the problem domain before drawing conclusions for a possible solution and IT architecture. The general project setup and interdisciplinary staffing was guided by working streams according to the tripartite of desirability, viability, and feasibility [4], thus, including domain experts, IT and technology experts, user researchers, business specialists, and a project lead. The project was divided into three main phases: Exploration, Alpha prototyping, and Market launch. TABLE III provides an overview of the project phases. In the following we describe the usage of DT for RE along the structure of (1)

objective, (2) activities, (3) roles, (4) outcome, and (5) conclusion for each phase.

TABLE III. PROJECT FALCON: TIMELINE AND GOALS PER STAGE

Project phase / Activities / Deliverables	FTE*
Phase 1: Exploration (08/2016-12/2017)	8
- 35 qualitative contextual interviews with potential users - 55 insights, 4 personae - 10 technology insights session with internal experts - 24 idea concepts - 4 value propositions and business model descriptions - 5-10 customer journeys for each value proposition - 12 contextual interviews with home owners to test customer journeys - focus group with 5 craftsmen to test customer journeys - market and competitor analysis, definition of strategic partners - identification of potential revenue models and first draft of business plan - technology screening for IT architecture, cost estimation for IT - 1 high-resolution mockup including 6 core functionalities	
Phase 2: Alpha prototyping (05/2017-11/2017)	15
- 21 user tests of (iterations of) the mockup of phase 1 - MVP defined and product alpha produced (9 functions identified) - business requirements based on advanced competitor analysis defined - software and system requirements for the platform are defined - quantitative study with 250 customers validate user requirements - Development of an alpha-version prototype in Scrum sprints (12) - Epics (7), user stories (>1000), mockup, and flow charts are defined	
Phase 3: Friendly User Test and Market launch (12/2017-03/2018)	22
- Ongoing scrum sprints (7) and functionality testing with users - Implementation of a prioritized scope in software - Friendly user test with 25 participants including testing of functional journey, usability, navigation, bug identification	

*Full-time Equivalent (FTE)

Phase 1 *Objective:* The goal of the exploration phase was to understand the problem and create a clear product vision.

Activities: The activities followed the DT process of empathize, define, ideate, prototype, test [18]. The team *empathized* by conducting contextual interviews with potential customers of the platform, e.g., homeowners, installers, craftsmen, and tenants to elicit their needs and requirements. The interviews addressed their experience in daily situations and motivations behind improving or renovating the house etc. From this research, the team derived needs and insights (*define*), e.g. "*Installers prefer to involve known craftsmen to build the PV system to avoid quality issues*." For a second round of interviews, the team created ideas and stimuli for interview partners, mainly home owners and craftsmen, to test first value propositions of the platform (*ideate, prototype, test*). This time, the questions focused more on potential features of the platform that could be further developed based on the testing results. For the most promising value propositions, customer journeys (in form of storyboards [18]) were developed, which considered activity and task flows as well as contextual usage of platform features. The creation of Personae for homeowners and installers guided the process of adjusting requirements and reflecting features when users were not around. In addition, market and competitor research helped to define and understand the requirements of the target market and the potential solution. Insight sessions with internal experts of the case company helped the team to gain knowledge about technological requirements. A sequence of framing and re-framing of the problem domain due to new learnings from interviews and testing sessions with customers resulted in a shift from the initial problem statement on PV&B to a more generalized home improvement platform idea. Eventually, all tested requirements and core functionalities were aggregated into a larger, more complex product scenario.

Roles: In this phase, regardless of the expertise and assigned work stream, each team member was involved in the same activities while undertaking the steps of the DT process to elicit needs and requirements from customers and gain the same level of empathy for the problem statement.

Outcome: The final deliverable was the specification of the elicited needs and requirements in form of (1) a mockup (InVision) that visualized 6 core functionalities of the intended digital software platform, (2) customer journeys specifying the experience and context of use of the platform, and (3) a video showing the defined Personae interacting with the platform.

Conclusion: In phase 1, DT is the leading process for all activities and team members. The clear process framework guides a deconstruction of the complexity of the initial problem statement with an iterative step-by-step approach. Accordingly, the requirements elicitation techniques (e.g., interviews, focus groups, throwaway prototyping) are put into a logical sequence of team-based efforts, regardless of the specific expertise of the individual team member, resulting in a common understanding for the problem domain and a product vision expressing the elicited requirements.

Phase 2 *Objective:* The goal of phase 2 was to deepen the problem understanding and to develop a functional alpha version prototype based on the vision of phase 1.

Activities: Scrum was the guiding process framework that enabled the entire team to work in sprints. While sprint 0 was used to build up the programming and operational infrastructure for the digital platform (3 weeks), the user research and business model team refined the outcome of phase 1 based on additional qualitative and quantitative research. Starting with sprint 1, the sprint cycle time was reduced to two weeks. In each sprint, interviews were conducted to prioritize functions and test usability and user experience of the mockup. The team leveraged several DT tools to enhance communication and ideation with stakeholders to advance the product vision (e.g., mood boards, ideation workshops with other domains, feedback capture grids). Based on the feedback, the team translated user requirements into epics and specified user stories, flow charts, and iterated versions of the mockup.

Roles: The team configuration was adapted according to the objectives of phase 2. Scrum master and development team were brought on board as well as additional capacity to work on the business model. Four project members from the exploration phase stayed in the project as part of the user research team. In this project phase, task distribution according to the dimensions desirability, viability, and feasibility was intensified, however,

user researchers and business developers was still one union acting as the product owner in a team-based effort.

Outcome: The phase goal was achieved, when the team showed the feasibility of the platform and confirmed real interest by users and customers.

Conclusion: As the focus shifts from understanding the problem to designing the solution, the guiding process model of DT makes way for the development focused approach of Scrum. Still, the toolbox of DT can easily integrated methods and tools into the flexible Scrum framework to elicit, refine, and specify requirements. Similar to phase 1, this task, i.e., the role of the product owner, is seen as a team-based approach to includes various perspectives in an interdisciplinary setup.

Phase 3 *Objective:* The objective of phase 3 is the market launch of the platform.

Activities: In phase 3, every activity is focused towards implementation and market launch. Scrum is the guiding development framework as the development continues. Enhanced priority is put on defining the go-to-market strategy and a suitable offering and partnering approach. A friendly user test with participants, who match the persona profiles, is conducted to test usability and eliminate bugs.

Roles: The software development team is extended and a clear split between the technical team, the business model stream, and the user research team is undertaken.

Outcome: Since the project is still running, we cannot provide the final outcome so far. A minimum viable product for a home improvement platform in market is to be expected.

Conclusion: Selected DT tools still provide methodological support (e.g., for testing and ideation), yet common agile development practices dominate. However, what we observe, is a manifestation of DT as a predominant human-centered mindset advancing the market launch.

B. What can be learned from DT for RE practices?

The case study was analyzed from the perspective of the challenges outlined in TAB. I. to draw first conclusions for the potential of DT for RE as done in Section V. A much-shortened version of our results is shown in TAB. IV. In the right column we show the evaluation of the learning potential of DT for RE either with a "–" (no potential), or a "+" (promising potential).

V. Discussion and Future Directions

The aim of our ongoing study is to investigate the use of DT for RE in software-intensive projects. This is a new area of exploration which has not been systematically investigated yet. Our study set out to provide a first impulse to enhance this understanding with empirical evidence. The findings suggest that DT has the potential to support current RE practices and vice versa. We do believe, that an integration of both approaches leverages a symbiotic relationship. As both topics, DT and RE are broad-ranging, multiple opportunities for further research can be identified. We now draw conclusions from our preliminary findings and discuss directions for future work.

TABLE IV. Role and Learning Potential of DT for RE

Challenge	Role of DT for RE	Potential
C1	DT has traceability difficulties. Elicited requirements are captured on Post-its or via prototypes, mainly in an unstructured manner, to advance team collaboration and process speed.	-
C2	DT faces similar challenges regarding the availability of customers. Yet, due to the process-orientation, customer interviews are relatively easy planned in advance. This might help to overcome this challenge to a certain degree.	+
C3	DT also neglects non-functional requirements such as security or performance requirements. However, it strongly enhances the priority of eliciting usability requirements.	+
C4	Due to short planning time, DT may also lead to inappropriate architecture. However, risk mitigation might be achieved through the thorough "upfront" approach to explore the problem domain which also includes the evaluation of suitable technologies.	-/+
C5	DT supports a team-based approach to RE. Thus, knowledge is more evenly distributed throughout the team. Different viewpoints foster a more comprehensive elicitation, more likely to expose tacit knowledge of stakeholders and team members.	+
C6	Similar to C4, DT may also struggle with imprecise effort estimates. Some risk mitigation might be achieved through the "upfront" approach that creates a clear product vision and scope.	-/+
C7	DT also faces difficulties in the prioritization of requirements. Based on our findings, we cannot propose any solutions stemming from DT for this challenge.	-

DT provides a structured process to requirements elicitation for wicked problems. We found DT particularly supportive in approaching wicked problems and turning these problems into well-defined problems that can be addressed with established RE practices. DT provides a guideline to support the practical application of methods that are commonly used in requirements elicitation. In addition, it fosters creativity through continuous re-framing of problem- and solution domain to find the best solution for the user. We see DT as an "extended arm" for RE to grasp wicked problems, while RE offers a strong integration framework for DT into the software development life cycle. In future work, we envision to develop a reference model combining both approaches, while taking the perspectives of DT as process, toolbox, and mindset into account. We also see potential to extend our research to development projects of safety-critical systems, in which DT might help to understand the actual needs upfront, i.e. make the problem addressable, while RE then provides the methods to design the proper system architecture, safety certification requirements and so on. We see the type of problem as the crucial indicator for which approach to leverage. For well-defined problems, i.e. when the real-world problem is known and the solution is clear, the use of the DT as a process might be ineffective and "over-engineered". Even so, we are convinced that an enhanced mindset for a more human-centered

way of RE proves beneficial either way and reduces the risk of deploying a system that does not meet its customers' needs.

DT leverages a team-based effort for requirements elicitation. Alternatively put, in DT the role of the product owner is performed by an interdisciplinary team. This results in a comprehensive requirements elicitation effort from different viewpoints to make better and more informed decisions at later stages in the development process. However, team-orientation is, like other agile approaches, highly dependent on the individual expertise of people, which makes it a critical success factor in the application of DT. Within the team, low-fidelity prototypes ease communication, also with customers, which helps to expose tacit knowledge of the former and latter. However, this action-oriented working mode also results in minimalistic documentation efforts, just like in common agile approaches. We find it necessary to look into these challenges to encourage a more suitable balance between the "rough and dirty," explorative working mode of DT and agile practices and the necessities for traceability and documentation.

DT emphasizes the elicitation of user requirements with a special focus on usability. Our findings suggest that DT puts priority on deriving a comprehensive user experience and, thus, on eliciting requirements in terms of usability, workflows tasks, and user interface. This speaks for the symbiosis between RE and DT, as we expect higher completeness of requirements when both methodologies are applied complementarily.

DT supports a seamless integration of upfront and concurrent RE practices. Our findings hint at a "morphing nature" of DT in software-intensive development projects, evolving from process-guidance, via toolbox support, to the manifestation of a human-centered mindset of the project team. When approaching wicked problems, DT starts with a structured, upfront RE approach to define a clear product vision. It, then, transforms into a loose bundle of tools and a mindset that, both, link well to common agile practices. Further attention should be paid to this phenomenon as we expect additional insights that might ease integration efforts of DT and RE.

It is certainly too early to consider DT a lasting contribution to RE, but, as our results indicate, it indeed has left first traces. However, due to its novelty, DT must account for a number of anomalies that are not clarified yet. As we observed challenges of DT, like lack of formalization, we plan to analyze these shortcomings. Here, we see potential to learn from the more mature discipline of RE. Particularly, we will continue to investigate phase 4 of our case, when the product is in market and new features need to be defined to extend the platform's core.

In general, we call for a joint support of the research community to help advance DT and RE, in the hope of conducting software-intensive projects in a more human-centered and effective way in the future.

REFERENCES

[1] Y. Yoo, "Design Thinking for IS Research," *MIS Quarterly*, vol. 4, no. 1, pp. iii–xviii, 2017.

[2] V. T. Heikkila, D. Damian, C. Lassenius, and M. Paasivaara, "A Mapping Study on Requirements Engineering in Agile Software Development," in *41st Euromicro Conf. on Softw. Eng. and Advanced Applications (SEAA'15)*, pp. 199–207, 2015.

[3] R. Buchanan, "Wicked Problems in Design Thinking," *Design Issues*, vol. 8, no. 2, pp. 5–21, 1992.

[4] T. Brown, "Design Thinking," *Harv. Bus. Rev.*, vol. 86, no. 6, pp. 84–92, 2008.

[5] J. Kolko, "Design Thinking Comes of Age," *Harv. Bus. Rev.*, Sep. 2015.

[6] R. Harte et al., "A Human-Centered Design Methodology to Enhance the Usability, HF, and User Experience of Connected Health Systems," *JMIR Human Factors*, (4:1), e8, 2017.

[7] S. Lohr, "IBM's Design-Centered Strategy to Set Free the Squares," *The New York Times*, date accessed 2 February 2018. https://www.nytimes.com/2015/11/15/business/ibms-design-centered-strategy-to-set-free-the-squares.html

[8] C. Vetterli, F. Uebernickel, W. Brenner, and C. Petrie, "How Deutsche Bank's IT Division Used Design Thinking to Achieve Customer Proximity," *MIS Quarterly Executive*, vol. 15, no. 1, pp. 37–53, 2016.

[9] M. Holloway, "How Tangible is your Strategy? How Design Thinking Can Turn Your Strategy into Reality," *Journal of Business Strategy*, vol. 30, no. 2, pp. 50–56, 2009.

[10] T. Beyhl and H. Giese. "Connecting Designing and Engineering Activities III," in *Design Thinking Research, Understanding Innovation*, H. Plattner, C. Meinel, and L. Leifer (eds), Cham: Springer-Verlag, pp. 265–290, 2016.

[11] C. Vetterli, W. Brenner, F. Uebernickel, C. Petrie, "From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking," *IEEE Internet Comp.*, vol. 17, no. 2, pp. 91–94, 2013.

[12] M.P. Soledade, R. Freitas, S.M. Peres, R. Steinbeck, U. Araújo, "Experimenting with Design Thinking in Requirements Refinement for a Learning Management System," in: *Anais do Simpósio Brasileiro de Sistemas de Informação*, pp. 1–13, 2013.

[13] P. Newman, M.A. Ferrario, W. Simm, S. Forshawz, A. Friday, J. Whittle, "The Role of Design Thinking and Physical Prototyping in Social Software Engineering," in *Proceedings of the 37th Int Conf on Soft Eng (ICSE)*, pp. 487–496, 2015.

[14] F. Häger, T. Kowark, J. Krüger, C. Vetterli, F. Uebernickel, M. Uflacker, "DT@Scrum: Integrating Design Thinking with Software Development Processes," in *Design Thinking Research: Understanding Innovation*, H. Plattner, C. Meinel, L. Leifer (eds), Cham: Springer-Verlag, pp. 263–289, 2015.

[15] L. Kimbell, "Rethinking Design Thinking: Part I," *Design and Culture*, vol. 3, no. 3, pp. 285–306, Nov. 2011.

[16] H. A. Simon, *The Sciences of the Artificial*, MIT Press, 1996.

[17] W. Brenner, F. Uebernickel, and T. Abrell, "Design Thinking as Mindset, Process, and Toolbox," in *Design Thinking for Innovation: Research and Practice*, W. Brenner and F. Uebernickel (eds), Cham: Springer, pp. 3–21, 2016.

[18] Bootcamp Bootleg d.school. *Stanford d.school at Stanford University*, date accessed 14 February 2018. https://dschool.stanford.edu/resources/the-bootcamp-bootleg

[19] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges," *Comput. Human Behav.*, vol. 51, pp. 915–929, Oct. 2015.

[20] D. Méndez Fernández and S. Wagner, "Naming the Pain in Requirements Engineering: A Design for a Global Family of Surveys and First Results from Germany," *arXiv [cs.SE]*, 2016.

[21] P. Runeson and M. Höst, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering, *Empir Software Eng*, vol. 14, no. 2, pp.131–164, 2009.

[22] R. K. Yin, *Applications of Case Study Research*. SAGE, 2011.