Designing Human-Centric Information Systems: Towards an Understanding of Challenges in Specifying Requirements within Design Thinking Projects

Jennifer Hehn[1], Falk Uebernickel[1], Emanuel Stoeckli[1], and Walter Brenner[1]

[1] University of St.Gallen, Institute of Information Management, St.Gallen, Switzerland
{jennifer.hehn,falk.uebernickel,emanuel.stoeckli,
walter.brenner}@unsig.ch

Abstract. Design-inspired approaches to problem-solving, like design thinking, have been gaining rising recognition as a way to create human-centric solutions. A design thinking project provides a guiding, yet not rigid framework to explore and specify requirements for the development of a new product or service. We investigate the challenges concerned with specifying requirements in design thinking projects with the purpose of laying the groundwork for creating solutions to these challenges. We find that design thinking supports the process of discovering requirements but leads to challenges stemming from the design thinking process. We identify conclusions for practice and suggestions for further research to create better human-centric information systems in the future.

Keywords: *design thinking, specifying requirements, human-centric IS*

1 Introduction

Digitalization is setting new agendas in almost every industry. The uncertainty in managing complex challenges within rapidly changing environments has prompted organizations to rely on human-centric and design-inspired approaches to create superior value for their customers. In this context, the notion of design thinking as a "way of finding human needs and creating new solutions using the tools and mindsets of design practitioners" [1:24] has become popular outside design professions and has left first traces in the digital and business world. Procter & Gamble, IBM, Deutsche Bank, and SAP internalized design thinking principles with proven success and are now considered poster children of the movement [2], [3], [4].

The guiding principle of design thinking – to create human-centric solutions that address real user needs – helps in the early stages of (new) product development to address "wicked problems" and develop a vision on how a solution should meet user needs [5]. Design thinking elicits those (hidden) needs and requirements, leveraging a holistic approach with a wide toolset of qualitative research methods and a series of low fidelity prototypes [6]. During a design thinking project, many digital and non-digital artefacts evolve. They express the identified user needs, insights, learnings, and design decisions. For reasons of speed, project teams capture and specify mentioned

Multikonferenz Wirtschaftsinformatik 2018,
March 06-09, 2018, Lüneburg, Germany

artefacts in a mainly unstructured form, e.g., through post-its, pictures, and low-fidelity prototypes during the process and text documentations on the occasion of specific milestones. This predominantly informal and unstructured mode of specifying requirements can be critical for two reasons which are strongly interlinked. First, it can hinder the team's ability to reflect and build on intermediate outputs and make decisions for further activities. Thus, it might result in a less desired prototype and, eventually, in a less desired product on market. Second, it also decreases the quality of the outcome when the final prototype is insufficient in explaining (1) certain design decisions made for the prototype and (2) the contextual details, in particular, when handed over to other teams for implementation (i.e., often software engineers) that are not able to revisit activities of the design thinking team. The lack of information might then lead to the creation of a less desirable product not fulfilling all relevant needs [7-9]. As a consequence, the innovative vision might not be able to develop its full potential because its realization does not conform to the elicited requirements of the design thinking team.

This study focuses on understanding the challenges of design thinking teams concerned with specifying requirements. It does not address the specific handover challenges from design thinking teams to software engineers, which have been discussed in prior research [7-9, 10], but rather focuses on the challenges within design thinking. The handover challenge can, of course, then (partially) be a result of the aforementioned one. We build our research on a multi-case study and analyse the data of five project teams. They approach a problem statement given by a corporate sponsor with the use of design thinking in a university context. We investigate the activities of specifying requirements to create solutions and the challenges that the teams face in doing so within the design thinking project setup. The paper is organized as follows: First, we establish the theoretical background of the study by drawing on literature about elements and methods considered essential in design thinking and their role in eliciting and specifying requirements for digital solutions. Next, we describe our research approach, case selection, data sources and collection, and data analysis. Then we report our findings. Finally, we provide suggestions for further research activities.

2 Theoretical Background: Design Thinking

Following the recent call of Yoo to "advance the intellectual foundation of design thinking" [11:v] for IS research, we aim to connect the key principles of design thinking (e.g. focus on user experience, future-orientation) to IS research. As such, we recognize the growing interest in design thinking as a subject for research.

Design thinking is best viewed from a trans-disciplinary perspective. The origins of design thinking date back to the late 1960s. Ever since then, design academics have been generating an extensive research history on mental processes that underlie design activities and their transformation into normative guidelines for creative problem solving [12]. These studies have been broadening the scope of design beyond the boundaries of product styling to a way of thinking that can be universalized for various disciplines. Looking at the user, use, and utility of a software product through the lens of design thinking can support IS development by enabling teams to invent new constructs and artefacts. As the quality of an information system can be judged according to the extent it meets its original purpose, human-centric design cannot only help to define the purpose but also its shape [6].

Design thinking process models, accompanied by a set of design tools, can provide a supportive framework for exactly this endeavor. Because of the clear instructions of those models, they are often utilized to conduct design thinking projects. This paper draws on a prescriptive process model that has been adapted from a design thinking course program at Stanford University (also called ME310). Figure 1 shows design thinking as a combination of a micro- and macro-process.

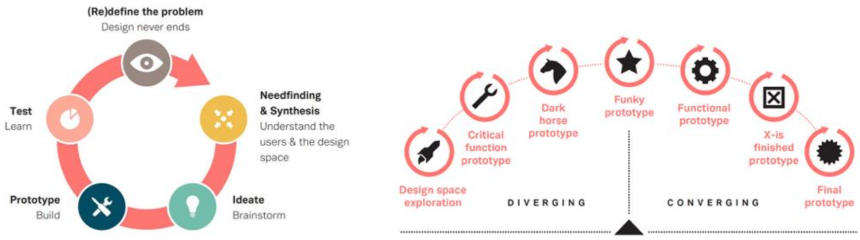

Figure 1. Design Thinking micro (left) and macro process (right), adapted from the course program at University of St.Gallen (http://dthsg.com, accessed on 24 September 2017)

The micro-process is integrated into a broader milestone model (i.e., the macro-process) and repeated in each phase. The macro-process includes an extensive pre-research phase (i.e., the design space exploration) and six different prototyping phases, which eventually merge to one final prototype. The creation of prototypes is not necessarily an expression of converging to a pre-defined solution from project start to end, it represents rather a diverging approach to iteratively question prior assumptions in order to create solutions that go beyond existing alternatives. In this respect, the mainly diverging character of design thinking differs significantly from the converging tendency of agile software development approaches. Iterations are carried out whenever and wherever necessary as they are essential to adapt to new information and gain deeper insights. Accordingly, the trial-and-error-principle of "fail early and often" is frequently adopted to encourage rapid learning from short project cycles [2].

Despite its guiding process model, the design thinking framework is flexible in terms of accommodating diverse perspectives and adapting to changing requirements. While methodological support on design thinking is widely available, there is a lack of guidance for effectively specifying and documenting artefacts and requirements in design thinking projects. This paper takes into consideration the following definition of requirements as "the descriptions of services that a software system must provide and the constraints under which it must operate." [13]. By doing so, requirements can be seen on different abstraction levels, from very high-level statements of service descriptions down to detailed specifications of certain functions or algorithms. There is a common understanding to differentiate requirements according to their nature. User requirements reflect the needs, desires, and stated functionalities of a software system.

System requirements express services needed from a technological view for the system to operate successfully. And lastly, software requirements specify the architecture of the software system in detail. On a more granular level, the requirements engineering community demands for requirements that are unambiguous, correct, complete, concise, feasible, understandable, and consistent [14].

The growing research effort in design thinking takes mainly two directions in the field of IS: (1) how IT and IS can support design thinking practices by generating new tools, and (2) how design thinking can support problem-solving in the IS context. The latter is subject of this study where related work is rather scarce. Prior research has acknowledged how design thinking can help in specifying requirements, mainly by conducting first small-scale experiments or cases in specific industries (e.g., [15], [16], [17]). For this study, results of research efforts at the Hasso-Plattner-Institute are particularly of interest. Häger et al. have shown that design thinking faces integration challenges when it comes to handing over prototypes to implementation and have called for an integrated approach of design thinking and Scrum to combine idea generation (design thinking) and product development [10]. In addition, Beyhl et al. have pointed out the challenge of documenting and traceability in design thinking projects, which is defined as "the ability to describe and follow the life of a requirements, in both a forwards and backwards direction" [18], [7-9]. Building on this knowledge base, we created our study. We adopt requirements specification as a key concept and explore how design thinking projects apply this for creating human-centric solutions. We look at five projects to identify challenges in the process of specifying requirements.

3 Research Context and Methodology

Our research adopts a multiple-case study approach and follows the guidelines of Yin [19] as it is particularly suitable for addressing the exploration of the research question: What are the challenges of specifying requirements within design thinking projects? The cases derive from a one-year university course as a part of which teams take on a design challenge provided by a corporate sponsor and guided by the design thinking framework introduced in Section 2. All corporate sponsors were either long-standing partners of the university or undertook a comprehensive onboarding process into the design thinking program to build up trust with the teaching team. We chose a multiple-case study approach as the cases we present are mainly replications of each other. This way, we expect better generalizable and more robust results for our findings [20]. We analysed five cases referring to five design thinking project teams (Table 1).

Table 1. Case details

Case	A	B	C	D	E
Team size	6	3	7	3	7
Corporate Sponsor	Alpha, Germany	Beta, France	Gamma, Germany	Delta, Germany	Epsilon, Colombia
Type of Industry	Pharma	Pharma	Pharma	Financial	Financial
# of employees	~ 50'000	~ 110'000	~ 48'000	~ 142'000	~ 50'000
# of prototypes	33	32	32	28	22
Final prototype	Chatbot	Smartwatch App and CRM Plug-in	Platform and Hardware	Smartphone App	Service, Smart-watch, and App

We selected five out of nine projects following the criterion that all the projects were addressing digital challenges. All teams consisted of business master students and, depending on the corporate challenge, included students from design, business informatics, and engineering. The students had no or little prior experience in applying the presented design thinking framework, however, they shared the experience of the same course content as a common background. To evaluate our research question, we collected multiple sources of evidence consisting of (1) project documentations, (2) physical artefacts, and (3) participant-observations (see Table 2).

Table 2. Sources of evidence (SoE)

Name	Type	Short Description	Number	Audience
Source of Evidence 1 (SoE1): Text and Visual Documentations				
One-page Summary (“One Pager”)	Text and pictures	Short description of project status after each milestone	6 per case	Corporate Liaison Teaching Team
Intermediate Documentation I	Text and pictures	Insights and learnings from first two milestones (35-120 pages)	1 per case	
Intermediate Documentation II	Text and pictures	Learnings from Darkhorse and Funky phases in the form of a booklet	1 per case	
Final Documentation	Text and pictures	Guidelines to understand the final prototype (40-140 pages)	1 per case	
Pictures	Photos	Photos of prototypes and other outputs	>1000 (all)	
Videos	Video	Video shows the final prototype	1 per case	
Source of Evidence 2 (SoE2): Physical Artefacts				
Prototypes	Physical and digital	Low to high resolution representations of ideas	147 (all cases)	Corporate Liaison
Source of Evidence 3 (SoE3): Participant Observations				
Weekly Review Meetings	Lecturer Notes	Weekly one hour review with each team on progress and content	28 per case	Design Thinking Teams
Weekly Lectures	Lecturer Notes	Weekly 2-3 hour lectures	32 (all cases)	
Regular Performance Reports	One page from lecturer	Assessment of each team with an evaluation of their performance	6 per case	

We examined project documentations (SoE1) with the help of document analysis which is the “systematic procedure for reviewing or evaluating documents—both printed and electronic” [21:27]. We chose this approach because our topic is concerned with specifying requirements and we see all documentation produced within the project as an indicator for evaluating our case question. Our available data has a broad coverage over the entire project time in multiple settings and scenarios, which makes analysing documents advantageous for our purpose [19]. Across all cases, we reviewed 45 documentations and more than 1000 pictures. We especially looked at the following questions: Which requirements are specified and how? How understandable and complete are they? Are context requirements captured? In addition, we leveraged prototypes as a complementary data source (SoE2). In design thinking, prototypes are per se expressions of elicited requirements and design decisions, which makes them a vital part in the evaluation within our research. To this purpose, we evaluated how and which requirements were expressed as prototypes. Third, we leveraged participant-observation to gain insights about possible challenges (SoE3). The data sources we used were mainly notes and observations from review meetings about process struggles, team dynamics, and content discussions. Three of the authors were lecturers and coaches in this course drawing from multiple years of experience in applying design thinking in university-level projects and corporate settings.

We chose to draw on multiple sources of evidence with the aim of data triangulation to support construct validity for our findings [19]. Subsequently, we followed a structured procedure for each source of evidence. We applied content analysis to identify findings and used cross-case synthesis to derive the challenges to specify. In particular, two of the authors started to derive the challenges independently from each other by examining the documents. They identified relevant passages and deduced them overarching themes in the topic of requirements specification. Then, the two engaged in discussions about the challenges to iterate and merge their first set of challenges. The third author acted as a sparring partner in the process of shaping the final set of challenges to provide quality assurance and objectivity. Figure 2 visualizes the above described procedure and provides the transition to the next chapter that will conclude the observed practices and challenges in capturing requirements.

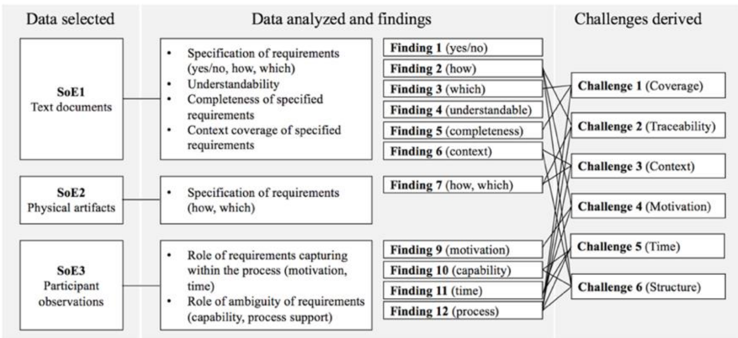

Figure 2. *Convergence of multiple sources of evidence (adapted from [19:117])*

4 Research Findings: Derived Challenges and Practices

4.1 General Findings

In all cases, the teams did capture and specify requirements throughout the course. Each team used different ways of specification all of which were generally understandable in form and expression. An explicit listing of requirements can be found in Case A and D, e.g., the final documentation of A states: *"requirements are derived from Alpha's initial brief, legal constraints in the industry, user interviews, Alpha staff interviews, and various kinds of testing"*. The requirements were structured into functional, experiential, and physical requirements. Each specification was further divided into (1) requirement, (2) metric, and (3) rationale as the example shows: (1) *"The solution must be available in the user's native language"*, (2) *"The solution should be translated to the native language of Alpha's main target markets,"* and (3) *"Our user tests showed* *us that patients have immense difficulty using a service if it's not offered* in the language *they're used to."* Furthermore, in Case D pictures of the prototype were used to show functionality and user requirements. The specification structure looks as follows: (1) issue description, (2) user quote, (3) need, (4) requirements derived. The other cases (B, C, E) use expressions like "features", "functions", "experiences" and "implications" to address the topic of requirements. In addition, prototypes served as a useful purpose to apprehend requirements. Namely, a prototype can take the form of a visual documentation and, thus, should be able to express requirements in a concise and understandable manner. Not all prototypes were self-explanatory though, and several needed further explanations to comprehend their purpose.

4.2 Output-Related Challenges

Challenge 1 (Coverage): Strong focus on user requirements while neglecting software and system requirements. The better and clearer the requirements the easier it is to implement them in the state of the intended purpose. During our analysis, we found a strong focus on specifying user requirements, which was one of our questions for SoE1 and SoE2 ("Which requirements are specified?"). Across all cases we see at least 80% of requirements addressing users, their needs, insights, and learnings from prototype testing. This finding is supported by notes from review meetings where discussions mainly revolved around user needs and how to transfer them into solutions. For example, a protocol of Case C includes the question *"How can we transfer the need for trustworthy information sources into a product feature? I don't know how to do that in the best way"*. Cases C and E address user requirements only, while documentations of Cases A, B, and D also show implications for software requirements, mainly in the appendix of such documentations. These made up the other 20% of the requirements documentation. The specification of system requirements was neglected completely. This finding is not surprising as the paradigm of human-centric design is the foundation of all activities in design thinking, putting the (potential) user in the role as a recurring sparring-partner for prototype testing. Still, feasibility and viability should be considered as well [2]. The lens of feasibility demands an exploration of organizational capabilities and technological options to translate the human-centric requirement into actual products and services. We see potential for design thinking to increase focus on this topic when evaluating prototype options for next steps in the process.

Challenge 2 (Traceability): Weak links between needs, insights, learnings, and requirements. Being able to trace back requirements to insights, needs, and learnings from testing helps to revisit previous decisions and to understand better the intended purpose of a solution. All sources of evidence (SoE1-SoE3) point out that achieving traceability seems to be a challenge in design thinking projects. Across all cases, the final documentations showed the most specific and traceable connection between need and (iterated) requirement, however, not always comprehensibly. We want to highlight the following evaluation notes from one lecturer in regard to the final documentation of Case E: *"The final prototype comes a bit out of the blue. [...] Which needs are addressed by which feature of the prototype is unclear."* Notes from review meetings show similar uncertainties when asking *"Where does this idea come from?"*, *"What is the need?"* (Case B, C). A good example provides Case A, which matched detailed requirements descriptions to a step by step journey of the user going through all screens of the prototype. Links to previous prototypes show the rationale for changing requirements. Case C provides another good example, where the team visualized the evolution from intermediate to final prototype for a better traceability between need and solution (Figure 3). The team added descriptions to explain which elements were transferred further and deducted implications for the final prototype. Traceability from final documentation to intermediate project results appears to be an important issue with varying results. We have seen good examples in our cases that can serve as benchmarks in the future. Nevertheless, notes from review meetings of the project have shown the teams' struggle and confusion in the attempt to recap and build on previous findings and specified requirements. We will pick this up again when discussing our observations for Challenge 6.

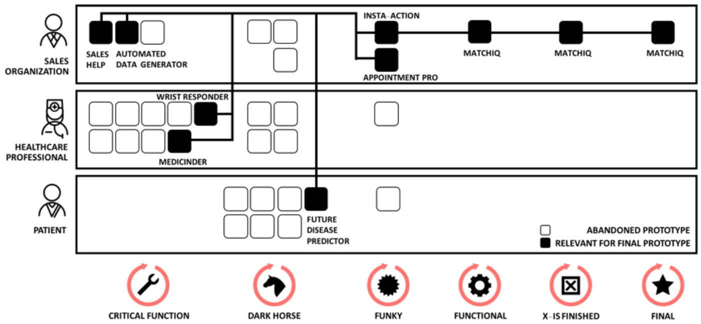

Figure 3. Evolution from intermediate prototypes to a final prototype (adapted from final documentation of Case C)

Challenge 3 (Context): Lack of formalizing context requirements. Contextual information and requirements help to create a holistic view of the entire systems landscape (e.g., in terms of constraints or system goals). Design thinking leverages a 360-degree approach (by multiple perspectives) to address the problem and solution space of the design challenge and, thus, aims to generate an integrated vision of the challenge and its context. Cases A and E used personas and empathy maps to set the user's context of the solution, e.g., in Case B, an element of the platform solution is a status function for doctors in which, similar to a frequent flyer program, they get points that translate into benefits. To put this into context, the team has specified the reason: *"we chose this solution according to the preferences of two target countries, where, unlike in Switzerland, the visit of sales representatives is appreciated as personal and individual attention of company Beta. Thus, novel digitization efforts are harder to argue. This is why we wanted to incentivize using our platform in a comprehensible manner"*. Despite design thinking being virtually predestined to elicit, and specify context requirements, we can determine a lack of formalized context requirements in all cases. A consistent way in specifying these requirements according to the context of the three dimensions of desirability, viability, and feasibility is lacking. This missing methodological support causes a lack of information in later implementation stages. Again, this finding can be related to the structural discussions in Challenge 6.

4.3 Process-Related Challenges

Challenge 4 (Motivation): Lack of motivation to specify requirements systematically. Although discovering needs and eliciting requirements is part of the design thinking process, documenting them in a systematic manner was found to be challenging. The deeply explorative approach of design thinking demands a multitude of sources and tools to create a multitude of artefacts, not yet knowing which ones will be the most relevant in later stages. But relying on implicit knowledge in a sea of analog and digital artefacts constitutes the risk of losing information. In general, the teams in our cases saw less of a contribution in specifying the requirements for themselves than for the teaching team and their corporate sponsor. The main motivational factor was the fact that a part of the grading in the course was based on the quality of the prototypes and their written deliverables. In addition, they also mentioned a feeling of obligation towards their corporate sponsor to specify findings for their later usage (A, B, C). To tackle this issue, three teams assigned a specific documentation role within their team. The teams that did not do this (B, E) showed particular weaknesses in the completeness and understandability of requirements specifications.

Challenge 5 (Time): Lack of time to specify requirements systematically. The element of available time is a crucial factor in all design thinking projects. Evidence for challenges in regard to time is mainly derived from weekly review meetings with the teams (Figure 2). One team member, for example, stated, *"the design thinking project is a marathon in the speed of a sprint"*. According to sources from review meetings (A-E), team members stated that they were aware that someone should document continuously, but failed to do so due to a permanent lack of time. The teams reported to prioritize time for *"real work like conducting user research or building the next prototype"* over specifying requirements in a systematic manner. This triggers a similar observation to Challenge 4 (motivation) as the task of specifying requirements is seen as a timely activity that gets de-prioritized under time pressure for *"more important things."* However, an indication that a more systematic approach could be beneficial, shows the following reflection statement from Case A: *"The fact that we had eight whole months to burn, often allowed us to slip into sluggishness, which really showed in our lackluster results in the winter period, and I feel that our indecisiveness cost us the chance to develop a proof of concept that actually validated ALL, not just most, of our claims and theories about the potential of chatbots in patient support"*.

Challenge 6 (Structure): Knowledge is implicit or captured on Post-its while adequate tool support is missing. In design thinking projects, there is no pre-defined structure on how, when, and which requirements to specify. Hands-on activities like building, testing, and experimenting are the preferred choices to gain insights and feedback from users. In all cases, observed practices for specifying requirements take a multitude of forms: working artefacts are mainly Post-its, sketches, pictures, low resolution prototypes, digital shared notes (e.g., on Trello), and a loose collection of word documents in a team share storage; official documentation purposes are usually addressed by natural language in text form, a high-fidelity prototype, and a final video to provide context. Creating prototypes is an evolving process and a way of "thinking with hands," where the team can reflect on an idea. However, often decisions on changes are not made explicit. This supports the finding that much knowledge stays implicit. Frameworks like personas, service blueprints, and empathy maps are commonly used to synthesize data from field research; adequate tools in the design thinking context for deriving and specifying requirements are lacking though. The "informal world of design thinkers" [7:167] (and, in the long-term, also the more formal world of software engineers) might benefit from a non-intrusive tool that fits to the "rough and dirty," explorative working mode of design thinkers. The combination of a prototype, showing links to previous prototypes and findings together with natural language specifications of requirements could serve as a first idea into this direction.

5 Discussion

Prior literature has acknowledged the challenges in specifying requirements in design thinking projects [7-9]. Our paper advances the understanding of this problem with data from a multi-case study of five design thinking projects. The results of our study show that some of the challenges can be considered as rather generic problems that also occur in other project-related teamwork (especially Challenges 4 and 5). The other challenges, however, suggest a strong association with the specific design thinking methodology. In this context, we find Challenge 1 as very IS-specific, whereas the other challenges (2, 3, 6) could also be transferred to other project setups that don't deal with creating and designing IS. Since we set out to create a holistic picture of the challenges to specify requirements in design thinking projects to create human-centric IS, we see all of our findings as relevant for scholars and practitioners alike. Practitioners can gain valuable insights to avoid or weaken aforementioned challenges, which can be influential for the success of a design thinking project. Academics benefit in many ways. First, our findings contribute to the body of knowledge for design thinking and IS research. Second, we believe that this paper will contribute to the understanding around aligning design thinking with later staged software development approaches. Third, Yoo suggests that "IS scholars can help design thinking practice by inventing new constructs, models, methods, and instantiations" [11:v]. Our research points to a specific area of future research, i.e., to enhance existing or creating new methods and tools for specifying requirements in such explorative approaches as design thinking.

Our paper is not without limitations. Yin stresses biased selectivity [19] as one flaw of analysing documents for research. Although it can be considered advantageous that our cases were conducted in the same setting, it can also be argued that the available documents are strongly aligned with the university's design thinking approach and its specific assessment policies for the course. In addition, the specific team staffing (students from different backgrounds and universities), the form of the corporate briefing, and the teaching team's coaching support should be considered as influencing factors on the derived challenges when transferring our findings to other (corporate) settings. Nevertheless, the entire course setup is crafted in a way that creates conditions that are as "real" as possible. Taking Sun and Kantor's [22] "three realities"-paradigm (real users, real problems, and real systems) for a naturalistic evaluation in IS projects as a reference, our course does address real problems from corporations (and not fictional ones) and applies real systems (i.e., the design thinking methodology with conducting real field research and user testing to create real solutions). We see "real users" as debatable, since our project teams are composed of students who can be more compared to external consultants than to internal project members of an organization. The latter ones for instance, often face internal politics or stakeholder management issues that we believe to be influencing design thinking projects immensely in "real corporate settings". Therefore, we encourage to analyse teams in actual organizational settings, allowing us to identify organizational specific challenges. Nevertheless, we still believe that the access to confidential data in the study outweighs the limitation because it led to a deeper understanding of the output- and process-related challenges.

6 Conclusion

We contribute to theory and practice on how to design and develop human-centric information systems by framing and discussing the problem of requirements specification in the context of design thinking projects. Although our results reflect the possibilities of design thinking to define human-centric solutions, our research reveals six challenges for specifying requirements, three of them related to the output of design thinking: (1) Design thinking teams focus on the specification of user requirements, while software and systems requirements are widely neglected; (2) the central outputs of design thinking projects (i.e., needs, insights, and requirements) are insufficiently linked with each other and, therefore, traceability is limited or even impossible for further use; and (3) the lack of formalized context requirements is limiting the use of project results in later implementation stages. Furthermore, we discovered three challenges relating to the process of design thinking itself: (4) Team members are hardly motivated to specify requirements systematically, (5) perceive the specification of requirements as an extra effort that shortens the already limited work time, and (6) lack in adequate tool support for specifying and structuring (implicit) team knowledge.

References

1. Kelley, T., Kelley, D.: Creative Confidence – Unleashing the creative Potential within us all, Crown Business, New York (2013)
2. Brown, T.: Design Thinking, Harvard Business Review 86 (6), 84–92 (2008)
3. Martin, R.: The Design of Business, Why Design Thinking is the Next Competitive Advantage, 1st ed., Boston (2009)
4. Vetterli, C., Brenner, W., Uebernickel, F., Berger, K.: Die Innovationsmethode Design Thinking. In: Lang, M., Amberg, M. (eds.), Dynamisches IT-Management. So Steigern Sie die Agilität, Flexibilität und Innovationskraft Ihrer IT, pp. 289–310. Symposion Publishing, Düsseldorf (2012)

5. Buchanan, R.: Wicked Problems in Design Thinking, Design Issues 8 (2), MIT Press, 5–21 (1992)
6. Vetterli, C., Brenner, W., Uebernickel, F., Petrie, C.: From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking, IEEE Internet Computing 17 (2), 91–94 (2013)
7. Beyhl, T., Berg, G., Giese, H.: Connecting Designing and Engineering Activities. In: Plattner, H. et al. (eds.): Design Thinking Research, Understanding Innovation, pp. 153–182. Springer International Publishing Switzerland (2014)
8. Beyhl, T., Giese, H.: Connecting Designing and Engineering Activities II. In: Plattner, H. et al. (eds.): Design Thinking Research, Understanding Innovation, pp. 211–239. Springer International Publishing Switzerland (2015)
9. Beyhl, T., Giese, H.: Connecting Designing and Engineering Activities III. In: Plattner, H. et al. (eds.): Design Thinking Research, Understanding Innovation, pp. 265–290. Springer International Publishing Switzerland (2016)
10. Haeger, F., Kowark, T., Krüger, J., Vetterli, C., Uebernickel, F., Uflacker, M.: DT@Scrum: Integrating Design Thinking with Software Development Processes. In: Plattner, H. et al. (eds.): Design Thinking Research, Understanding Innovation, pp. 263–289. Springer International Publishing Switzerland (2015)
11. Yoo, Y.: Design Thinking for IS Research, in: Editor's Comments: Diversity of Design Science Research, MIS Quarterly 41 (1), iii–xviii (2017)
12. Simon, H.A.: The Sciences of the Artificial, Cambridge, London (1972)
13. CS Software Engineering Note 2: Software Requirements. http://www.inf.ed.ac.uk/teaching/courses/cs2/LectureNotes/CS2Ah/SoftEng/se02.pdf (Accessed: 23.09.2017)
14. Bahill, A.T., Madni, A.M.: Discovering System Requirements. In: Bahill, A.T., Madni, A.M. (eds.): Tradeoff Decisions in System Design, pp. 373–457. Springer, Cham (2017)
15. Newman, P., Ferrarioy M.A., Simm, W., Forshawz, S., Friday, A. Whittle, J.: The Role of Design Thinking and Physical Prototyping. In Proceedings of the 37th IEEE International Conference on Software Engineering, IEEE Press, New York (2015)
16. Harte, R., Glynn, L. Rodriguez-Molinero, A., Baker, P., Schwarf, T., Quinlan, L.R., Glaighin, G.: A Human-Centered Design Methodology to Enhance the Usability, Human Factors, and User Experience of Connected Health Systems: A Three-Phase Methodology, JMIR Human Factors 4, (1), 91–94 (2017)
17. Soledade, R., Freitas, S., Peres, M., Fantinato, R., Steinbeck, U., Araújo, F.: Experimenting with Design Thinking in Requirements Refinement for a Learning Management System, Anais do Simpósio Brasileiro de Sistemas de Informação, 182–193 (2013)
18. Gotel, O., Finkelstein, F.: Extended Requirements Traceability: Results of an Industrial Case Study. In: Proceedings of the third IEEE International Conference on Requirements Engineering, pp. 169–178. IEEE Press, New York (1997)
19. Yin, R.K.: Case Study Research: Design and Methods, 5th ed., Sage, Los Angeles (2014)
20. Eisenhardt, K.M.: Building Theories from Case-Study Research, Academy of Management Review 14 (4), 532–550 (1989)
21. Bowen, G.A.: Document Analysis as a Qualitative Research Method, Qualitative Research Journal 9 (2), 27–40 (2009)
22. Sun, Y., Kantor, P.B.: Cross-Evaluation: A New Model for Information System Evaluation, Journal of the American Society for Information Science and Technology 57 (5), 614–628 (2006)