

# The Use of Design Thinking for Requirements Engineering: An Ongoing Case Study in the Field of Innovative Software-Intensive Systems

Presentation of work originally published in the Proceedings of the 26th IEEE International Requirements Engineering Conference, Banff, Canada 2018

Jennifer Hehn<sup>1</sup>, Falk Uebernickel<sup>2</sup>

**Keywords:** Design Thinking, Requirements Engineering, Innovative Software-Intensive Systems

## 1 Extended Abstract

Effective Requirements Engineering (RE) is recognized to be one of the most crucial activities in software-intensive development projects. However, practitioners and scholars have also revealed its numerous challenges. Especially nowadays, when software development increasingly calls for agile and human-centered practices to address the often fuzzy needs of the various stakeholders involved. The popular approach of Design Thinking (DT) has gained recognition as a way to approach product and software development with an interdisciplinary team, qualitative user research methods, rapid (non-technical) prototyping techniques, and iterative learning cycles. This diverging way of problem-solving is notably different from the rather converging and more formal RE practices. We postulate that the strongly human-oriented working mode of DT can complement the more formal, technology-driven RE practices. This is a new field of exploration that has not been systematically examined yet. To enhance this understanding with empirical evidence we set up a longitudinal case study in an agile development setting from idea conceptualization to market-ready implementation. We investigated a software-intensive development project over a time frame of 1.5 years in a large utility company in Europe. The objective of the project was to design a digital platform in the energy sector. The project management decided to apply DT and (later) Scrum to better understand the problem domain before drawing conclusions too early. The interdisciplinary project team included domain experts, user researchers, business specialists, IT and technology experts, and a project lead. Table 1 shows the main project phases of (1) exploration, (2) alpha prototyping, and (3) market launch and their respective objectives, activities, roles, and outcomes.

---

<sup>1</sup> University of St.Gallen, Institute of Information Management, Müller-Friedberg-Strasse 8, 9000 St.Gallen, jennifer.hehn@unisg.ch

<sup>2</sup> University of St.Gallen, Institute of Information Management, Müller-Friedberg-Strasse 8, 9000 St.Gallen, falk.uebernickel@unisg.ch

<i>Phase</i>	<i>Exploration</i>	<i>Alpha Prototyping</i>	<i>Market Launch</i>
Duration	3 months, 8 FTE	7 months, 15 FTE	4 months, 22 FTE
Objective	Understand the problem and create a product vision	Develop a functional alpha prototype	Market launch of the platform
Activities	DT as guiding process: (1) empathize, (2) define, (3) ideate, (4) prototype, and (5) test in several iterations	Scrum as guiding framework; DT tools to enhance communication and ideation with stakeholders	Scrum as guiding framework; enhanced priority on de-fining the go-to-market strategy; friendly user tests
Roles	Each team member is involved into all activities to elicit needs and requirements	Onboarding of development team; business model focus; DT team transitions into Product Owner role	Software development team is extended; split between technical, business model, and Product Owner role
Outcome	Mockup with core functionalities; customer journeys define context of use	Proof of Concept demonstrates feasibility and viability of the solution	Minimum Viable Product (MVP) is ready for market entry
Conclusion	DT as guiding process; requirements elicitation is a sequence of team-based efforts	DT used as a toolbox; Scrum is guiding framework; part of DT team becomes Product Owner	DT as a mindset; Selected DT tools provide support, yet agile development practices dominate

Tab. 1: Project Overview (adapted from [HU18])

We found that DT has the ability to address some of the known challenges in agile RE, e.g., to expose tacit knowledge of stakeholders through prototypes. In particular, we learned these lessons by using DT to elicit requirements and define a platform solution:

- *DT provides a structured process to requirements elicitation for wicked problems.* DT offers a prescriptive guideline to apply methods which are commonly used in RE to elicit stakeholder needs and requirements.
- *DT leverages a team-based effort for requirements elicitation.* The role of the Product Owner is inhabited by an interdisciplinary DT team leading to a comprehensive requirements elicitation through various viewpoints.
- *DT emphasizes the elicitation of user requirements with a special focus on usability.* DT puts priority on discovering requirements related to usability, workflows tasks, and user interface.
- *DT supports a seamless integration of upfront and concurrent RE practices.* DT evolves from an upfront definition of stakeholder needs, into tool support with human-centered principles that, both, link well to common agile practices.

DT offers several possibilities for coping with today’s (agile) RE challenges. We see potential to combine DT and RE in many dimensions to help create human-centered software solutions more effectively. We consider DT as an “extended arm” for RE to approach wicked problems and explore actual needs upfront with a prescriptive process, while RE provides an integration framework for DT into later staged software development life cycles. However, in our study we also found that DT must also account for a number of anomalies that are not clarified yet, similar to agile practices. We plan to analyze challenges like the high dependency of people, traceability difficulties, or the lack of formalization. Here, we plan to learn from the more mature discipline of RE.