A designerly methodology for software development

Background to research: Trends influencing industrial product development

The knowledge platform of this promotion research is industrial product development that has been strongly influenced by four major trends in recent decades. These trends are: (i) diversification of human and social needs, (ii) rapid development and uncontrolled proliferation of advanced technologies. As a reaction upon these, there is an (iii) evolution of product manifestations and implementations, and consequently there is a need for (iv) increasingly sophisticated design approaches and methodologies. Together, these trends have led to a characteristic shift in the type of products and the approaches to product realization. Fuelled by the developments of technologies, new product paradigms have formed, nowadays, three generations of products can be identified: The first-generation products are assembled hardware products (e.g. bike), software implementations (e.g. Word), and pure services (e.g. child care). The second-generation products show a growing level of integration and move towards integrated systems. They are substantiated by three types of systems: (i) product-service systems (e.g. car sharing), (ii) embedded systems (e.g. washing machine), and (iii) information systems (e.g. stock management system). The third-generation products are complex systems and environments, which are conceptualized and implemented according to the principles of cyber-physical systems (e.g. intelligent traffic system). The evolution of products is also reflected in the shift of designers’ focus, as their attention is moving away from pure form, function, materials and manner of production concerns and towards utility, usability, human experience and desirability concerns. Accordingly, the meaning of the products is a more and more important phenomenon for designers. The shift in design approaches from function-focused ones through consumer-oriented ones to human-centered ones is an essential development. This provided the starting knowledge platform and motivation for this promotion research, and played an important role in conceptualization of the research problem.

Research domain

In this research we have concentrated on the development of software as products or components of complex systems. This problem domain was chosen because software (i) yields the largest opportunities to meet the requirements rooted in complexity and evolution, and (ii) has a large influence on the sophistication of products, but (iii) is also the most difficult part to develop in complex systems. In the last decades, there was an intense diversification of software products and continuation of this process can still be observed. Software products manifest as self-contained application packages, but also as embedded software for controlling systems, or agents of complex information systems, or synergetic constituents of cyber-physical systems.

For this research, the type of software we specifically focused on was defined by acknowledging a real-life need, namely, that of an interactive and knowledge-intensive design support tool able to facilitate concept generation and trade-off forecasting in case of ubiquitous augmentation of domestic appliances. A software tool providing the necessary functionality for this reason was selected as a test case for our human-centered software development methodology. This test case also took the role of an archetype of a family of similar design support tools. By using it as a reference case, we could consider a family of design support tools in our work and grasp a range of technical and human issues associated with the dedicated software development methodology. The novelty of the reported research approach is in the epistemological, methodological and procedural symbiosis of the methodology development and the application case development. It was assumed and has been confirmed that the dialectic interaction of the support tool and the application case provides benefits for both.

Research problem and objectives

This PhD research strongly envisioned that software tools belonging to the category of interactive application software (e.g. design support tools) should be developed according to a participatory design strategy. As the interacting trends imply, the close relationship between products and their users creates the need for a more intense stakeholder involvement during the design process. This enables designers and engineers to understand the needs for change and to cope with the challenges of complex functionalities and fast realization processes of second and third generation
products. Our observation was that the human-centered design of interactive software has not reached the desired and potential level, compared with many hardware products. The research vision was that a designerly methodology was needed in order to solve the above-mentioned problem of interactive software development. As a research problem this poses two challenges: (i) reconceptualization of the development process of interactive software towards a designerly (stakeholder-oriented) approach, and (ii) establishing a robust basis for a new methodology that covers the early phases of software development where critical decisions are made. Our primary objective was not increasing the efficiency of the product development, but increasing the utility and quality of interactive software products. By involving the stakeholders in the early phases, software products can be more custom-made and more needs’ appropriate. Despite the extra time and efforts, it is worth involving stakeholders in the utility and quality enhancement of software.

Consequently, the objective of the PhD research was set to conceptualize, elaborate and test a designerly software development methodology (DSDM) that supports stakeholder involvement in the most critical phases of software design. We decided to apply a structured view on the software development process and introduced a methodological framing by which we could focus on the subsequent phases. Stakeholder involvement has to start during the identification process of the design requirements and when an overall conceptual framework of the software tool is constructed. Stakeholders should also be involved when the concept of the software tool has been developed (it should be demonstrated to stakeholders and justified and validated through their involvement). Finally, stakeholders should be involved when a pre-implementation version is completed and take part in the testing and critique of this. To complete these activities efficiently, the above phases need dedicated methodologies that we called single-phase (component) methodologies. They were coherently and transitively integrated into the targeted multi-phase support methodology, called DSDM.

Research hypothesis

Focusing on humans and their experiences is a key-issue in current product development. Our generic research hypothesis suggests that software development could benefit from following the principles of human-centeredness that are applied in traditional product development. Based on our previous literature study and practical experiences, we investigated the differences between the development of hardware and software products. Furthermore, we investigated why we cannot directly use the human-centered design principles of consumer durables to software development. Our generic research hypothesis claims that specific methodological principles gathered from the domain of modern consumer durable development could be used as a basis of the targeted designerly software development methodology. There is extensive literature on the principles and approaches of human centeredness in consumer durables design, where optimal physical and cognitive interaction with humans is an important factor of product success. In this domain, designers have a firm intention to customize the product to end users and, towards this end; they closely involve and interact with various stakeholders in the development process. The stakeholder involvement is supported by the use of various demonstration means, visuals, and virtual and physical prototypes, such as sketches, mock ups, CAD models, and tangible prototypes. In the design process, prototypes are used to discuss and evaluate the design with stakeholders towards improvements. Verifications and validations occur at different phases of the development process and consequently different means are used. Incorporating the relevant principles of consumer durables development into the domain of interactive design software development is, however, not straightforward. There are important differences between the two domains. The most significant ones are: (i) the difference in the tangibility or material manifestations of products, which entail different prototyping means, and (ii) the difference in the interaction with the physical product and software products. It seems that it is more difficult to actualize early demonstrations of intangible products and consequently, they require a higher level of empathy from stakeholders with the design, as well as an ability to provide suggestions for improvements.

Overall research approach

Due to the variety of objectives and contexts, a multi-methodological framing was applied to set up the research design. The whole of the PhD research was broken down into five interrelated research cycles (RC). Each cycle had its own objectives, context, and framing methodology. For this purpose, the methodological framing theory, proposed by Horváth, has been applied. The objective of the research cycles was to investigate the needs of the specific phases of the idealized multi-phase process (framework and requirement aggregation, concept development, and system elaboration). In the first research cycle we investigated the need for stakeholder involvement in the current software development approaches, and described the context of the research process. We analyzed the phenomena of stakeholder-oriented design, and considered the gaps and important issues to deal with in our methodology. During the execution of RC 2, 3 and 4, we investigated the three most critical phases discussed above. In research cycle 2 we examined the methodology development issue in the context of requirements engineering and framework ideation. In research cycle 3, the context and the influencing factors of enabling concept synthesis and demonstration were investigated. In research cycle 4, the research work focused on surrogate-based prototyping in the context of detailing functionality and usability testing. In
research cycle 5, we drew conclusions about the entire research through a multi-aspect external validation of the proposed multi-phase methodology.

**The essence of the designerly software development methodology**

Implementation of the working hypothesis led us to a theory that explained what kind of designerly software development methodology could support stakeholder involvement in the most critical phases of software design, and how this methodology could be applied with success. The *objective of the DSDM* is to systematize and facilitate the involvement of the stakeholders relevant to each phase of the development. DSDM was developed to support the implementation of second and third generation software products, in particular design support tools. These interactive software tools feature functional and structural complexity and the need for an ability to evolve in order to continue fulfilling the needs of their stakeholders. The practical aim of the proposed methodology is to achieve an optimal stakeholder involvement with the hope of increasing the efficiency and effectiveness of conceptualization and detail design.

Based on literature, we learned that a methodology should be specified by an underpinning theory and should offer procedural scenarios, problem solving instruments, a set of methods, and should define the criteria of goodness. The underpinning theory of DSDM formulates three conceptual pillars: (i) context-sensitive stakeholder involvement, (ii) managing complexity and evolvability, and (iii) achieving an increasing level of fidelity. In the application of the methodology, a key issue is to obtain constructive feedback from the stakeholders, including qualitative change and improvement proposals and/or quantitative measures. To handle the complexities accompanying comprehensive systems, we build on the principle of separation of concerns. The complexity of the targeted products also implied that it was not possible to consider all required possibilities, and the characteristics of the different stakeholders could not be known at the start of the development process. The outcome of the application of the component methodologies included in the DSDM operates with an increasing level of fidelity. This level depends on the amount of information available in the above three phases. The methodology we have developed suggests, starting with a high-level abstraction (that can be embedded in a low-fidelity prototype) and ending with a high-fidelity testable prototype of the detailed software system. These subsequent forms of prototypes can be adjusted to the contents, stakeholders and contexts. In the process of exploring the stakeholders’ opinion, ideas, and recommendations, the prototypes of a growing fidelity support both the interrogation and the constructive activities.

As mentioned above, the DSDM focuses on three critical phases of the software development process: (i) framework ideation, (ii) concept integration, and (iii) system development. Because of the diversity of the stages of the development process, different phase-methodologies have been proposed for each individual phase. In order to achieve an optimum support and efficiency, the proposed DSDM has been developed as a multi-phase stakeholder-oriented designerly methodology, which offers suitable procedures, instruments and methods for three single-phase methodologies: (i) critical collective reflection, (ii) modular abstract prototyping, and (iii) surrogate-based prototyping:

- The critical collective reflection (CCR) methodology supports stakeholders’ reflections on the requirements and the conceptual framework. By contrasting the proposals of the development team with that of the expert stakeholders contributes better framework solutions can be achieved. The methodology of CCR facilitates improved collective requirements engineering and framework conceptualization through the direct reflection of expert-stakeholders on the proposal demonstrated by the software developers.
- The modular abstract prototyping (MAP) supports concept integration and consolidation. MAP offers the possibility for a rapid development of modularly configurable and presentable content. It also supports focused demonstration to stakeholder groups and their decision making process, by using abstract prototypes. The proposed modular approach allows for the consideration of the differences in interests and viewpoints of the stakeholders and to demonstrate the software concept according to their needs. The discussion enabled by MAP will result in necessary change proposals and more consistent software concepts.
- The surrogate-based prototyping (SBP) methodology supports fast realization and investigation of testable, tangible prototypes, which are developed by using a composition of existing software components or platforms. Based on SBP, not only the functionality, but also the usability of the software products can be tested rapidly and at a low cost. As demonstration and testing means, SBPs facilitate the provision of concrete low-level feedback on the attributes and behavior of the software before its final realization. By using platform-based SBP, it is possible to go beyond the conventional concept of pure component-based design. A platform-based SBP reduces the problem of interfacing of heterogeneous components.

To conclude, the whole DSDM proved to be a valid methodology for the development of interactive software products that have complex functionality, user requirements that are rather uncertain and unclear in the beginning of the process, and that are complex due to environmental aspects, required modeling and data processing.