

Software Tools for Supporting Reflection in Design Thinking Projects

Thorsten Schoormann
University of Hildesheim
thorsten.schoormann@uni-hildesheim.de

Julien Hofer
University of Hildesheim
julien.hofer@uni-hildesheim.de

Ralf Knackstedt
University of Hildesheim
ralf.knackstedt@uni-hildesheim.de

Abstract

In creative work such as design thinking projects, teams mostly seek to solve complex (wicked) problems as well as situations of uncertainty and value conflicts. To design solutions that cope with these aspects, teams usually start doing something, reflect on their results, and adjust their process. By actually doing something, tacit knowledge (i.e., knowing-in-action) of individuals is disclosed, which might be beneficial for an entire project team because it allows drawing on information and experiences that go beyond single individuals. Accordingly, the present study aims to investigate how tools can be designed that support collaborative reflection in creativity-driven projects. Drawing on reflection theory and several expert interviews, we derive design requirements as well as present a concrete software-based prototype as an expository instantiation.

1. Introduction

Rapid changing environments and increasing uncertainty in markets are just a few examples of essential challenges for businesses in these days. In order to face these challenges, the development of innovative solutions is becoming a critical factor for organizations [1], [2]. Hence, there is a booming interest from both researchers and practitioners alike to explore and plan future endeavors in terms of performing creative projects that aim to ensure sustainable success in a dynamic world [3]. In addition to the need for being creative, also the process of creative work itself is changing. Due to the globalization and decentralization of workforces, for instance, the way people generate ideas, share experiences about innovations, or evaluate and discuss a set of proposed solutions is fundamentally affected. By employing new information and communication technologies, the collaboration of team members that are dispersed around the world can be supported [4]. Doing this, various actors are enabled to participate in

creativity-related activities such as develop new products, services, or even entire business models.

One of the prevailing forms in research and practice that aims to support creativity within organizations is design thinking [5-7]. In general, design thinking can be used to design solutions for complex problems, so-called wicked problems, that are hard to solve due to ill-defined, incomplete, conflicting, and changing requirements [8]. For applying design thinking, different mindsets including procedure models and methods have been proposed (e.g., [9]–[11]). However, regardless which creativity approach is used, finding solutions that address those complex problems is a challenging task because people cannot make sense of such problems in a complete manner and cannot consider all consequences that might occur by certain actions they plan to do. One way to cope with situations of uncertainty and value conflicts is to actually start doing something, think about what and how it happened, and adjust the doings. This intuitive process of reflection discloses tacit knowledge of a team to solve a problem (knowing-in-action) [12], [13].

Referring to design activities in particular, reflection is an essential activity. According to Schön [14], designers respond to demands and possibilities of designing something by continuous reflection and interaction with other individuals. In doing this, they make use of their own experiences (e.g., in the form of previous prototypes and solutions) and seek to adapt them to the current situation. As a result, conclusions to plan actions for future efforts are derived [15]. Moreover, conscious reflection on the problem framing and the emerging ensemble allows the research process to be adjusted based on lessons learned [16]. This especially applies to creative work as, for instance, complex problems need to be iteratively reframed to be more precisely, the process has to be adjusted after evaluations and feedback, and assumptions have to be verified. Accordingly, we argue that continuous and systematic reflection in and on such collaborative projects is fruitful and should be considered in future research on creative work.

While reflection is primarily seen as an individual activity, some authors emphasize the strong social

dimension [17]. If people reflect together, entire teams can be inspired by experiences from others [18], learn from others, obtain feedback, and create ideas that go beyond knowledge of single individuals [15]. To release such benefits, knowledge and experiences need to be shared among a group [19]. Due to the fact that collaborative reflection is assumed as valuable for coping complex problems and uncertain situations, this study aims to investigate how software support for design thinking can be designed to leverage reflection. Doing this, we enable users to learn from experiences and knowledge that already exists within a team. Therefore, we raise the following research question:

RQ: *How to design tools that support continuous reflection in design thinking projects?*

In order to answer this question, we conduct a design science study in which we iteratively build and evaluate a prototype for software-supported reflection in design thinking as well as seek to abstract knowledge about the class of such tools. We aim to contribute to the features that should be implemented to enable the entire progress of a creative project and the outcomes to be reflected. The solution can be translated into other solution spaces (e.g., IT-artefacts such as visual tools), and if implemented, help practitioners to collaboratively reflect their process and results. For academics, our findings may act as a foundation for advancing creativity research on software-supported reflection.

To pursue this goal, we first outline the research background on design thinking and reflection theory (Section 2). Based on the research method [20], we identify design requirements from theory, related work, and expert interviews (Section 3). Afterward, we describe the derived requirements (Section 4), specify design features, and instantiate them through a software prototype (Section 5). For evaluation, we demonstrate the applicability of our prototype by employing it in a design thinking project and analyzed log files from the software to gain insights in terms of usage (Section 6). Finally, we discuss our findings and limitations and conclude (Section 7).

2. Research background

2.1. Design thinking

Design thinking is employed in several disciplines such as product design [21], architecture [22], and management [6], [7]. It is a mindset that is human-centered because of the importance of empathy and human needs represent one central aspect in design thinking [1]. People's needs and desires are the sources

of inspiration—they generate insights, which forms an indispensable basis for creating new ideas [23].

Procedure models. The design thinking process is iterative, and thus, designers explore a problem by generating and testing various solutions while constantly reflecting in-depth problem characteristics, insights, ideas, and solution concepts [24]. As work within (usually interdisciplinary) teams aims to solve complex problems, collaboration is highly important [24]. This is emphasized because 'out-of-the-box' ideas raise from combining own thoughts and experiences with others. These ideas are not a result of one genius, but from collaborative work [7]. In order to support the application of design thinking, different mindsets including procedure models and methods have been developed in research and practice. While, for example, HPI D-School differentiates between six phases, namely *understand*, *observe*, *define point of view*, *generate ideas*, *develop prototypes*, and *test* [11], the process from Stanford d.school contains five steps for *empathize*, *define*, *ideate*, *prototype*, and *test* [10]. However, as a minimal consensus, these procedures mostly emphasize iterative designs, user-centered solutions as well as shifting between divergent and convergent thinking [2]. Thus, goals and constraints are constantly being reframed and solutions can emerge over time [21], [24].

Virtual design thinking. As design thinking is usually carried out in an analog setting, an emerging stream of research deals with the adequate use of IT/IS. Lattemann et al. [4], for instance, explored the support of tools such as digital whiteboards and file-sharing systems, and Redlich et al. [25] compared analog settings with semi-virtual settings and pointed out that there are no negative effects of virtual projects.

2.2. Reflection theory

Reflection has been endorsed as a practical approach in different sectors such as education, cooperative work, problem-solving, design and engineering, and learning (e.g., [15], [26], [27]). Reflective thinking is especially important when a team has to deal with unsolved complex problems [18] or confused and uncertain situations [28], which both apply to the context of design thinking projects.

Although reflection is mostly seen as a fuzzy concept [17], it generally contains activities for collecting experiences, re-assessing them in the faced situation, and deriving learnings for future actions [29]. According to Grushka et al. [30], hypotheses are formulated, a stand is taken, and a plan of actions is derived. Thus, reflection enables assumptions to be examined and assessed [31]. Referring to this study's purpose, to perform design thinking, the activities from

reflection are especially useful, because based on the formulation of a certain problem, different ideas and prototypes are created (i.e., hypothesis of ways to solve a problem), and feedback is collected across the entire team to gain potential for refinement. Consequently, we consider reflection theory to understand how experiences and knowledge can be reflected to derive learnings for the future in design thinking projects.

Types of reflection. Reflection is usually classified into two types [12], [32]: ‘*reflection-in-action*’ (i.e., assumptions and alternatives are evaluated during the action) and ‘*reflection-on-action*’ (i.e., retrospective analysis of actions and their effects). Schön stated that individuals often employ implicit knowledge in processes where they face situations of uncertainty, instability, and value conflict [13] [26].

Collaborative reflection. In general, reflection theory has some overlaps with other theories such as sensemaking, as it seeks to understand the past in a collaborative manner. However, a unique characteristic of reflection is that it has a strong focus on deriving insights for the future [34], which is important to guide actions related to the development of new solutions. Reflection is a part of individual and cooperative work aiming at guiding decision-making [29]. As reflection within a team is beneficial, for instance, to share experiences from different team members, learn from others, provide feedback, and develop solutions that go beyond ideas of single team members (i.e., cross-boundary) [15], [18], collaboration plays an essential role. Hence, design thinking teams need to enable different perspectives to be taken, knowledge and experiences to be disclosed, and communication to be made. In contrast, some shortcomings occur that especially deal with time (i.e., reflecting might take longer) and complexity (i.e., people need to be coordinated) [19], [31].

Software supported reflection. To overcome these shortcomings, software support can be applied which, for instance, facilitates actions such as sharing data, reflecting over a period of time, motivating activities, merging and linking experiences, obtaining feedback [19], [35], as well as remembering past situations, documenting outcomes [15].

Even though software can elevate the reflection in teams, current solutions mostly deal with individual reflection or on general collaboration without concrete features for reflection [19]. To the best of our knowledge, only limited studies seek to investigate how software support should actually look like. For instance, Prilla [19] propose two main features, namely (1) communities to enable discussion among a group of users and (2) promptings to trigger the participation of collaborative reflection. As another example, Renner et al. [15] derive potentials and prototypes that facilitate

reflection at work and Renner et al. [17] show how tools can be employed for reflective learning. Nonetheless, software support for reflection seems to be in an emerging stage and should be investigated in future studies that, for instance, draw on a theoretical-grounding of such tools. Moreover, how to support reflection in creativity projects is virtually neglected, which also presumes more research that aims to identify which features should be provided to best possible support collaborative reflection.

3. Research method

The primary goal of this study is to derive design requirements for the class of software-based visual design thinking tools that enable continuous reflection of design outcomes and the progress. Following the method as proposed by Kuechler and Vaishnavi [20], we ran through three major design cycles of our software artifact that serves as an instantiation of the derived design requirements (see Figure 1).

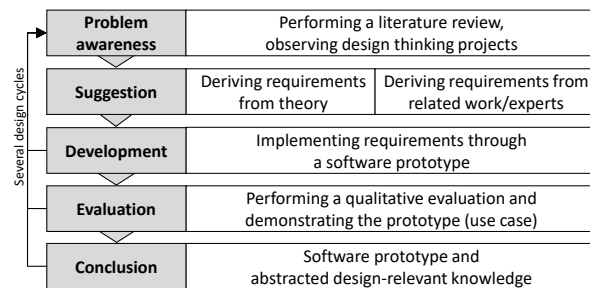


Figure 1. Design science research method

3.1. Design cycle 1 (initial design)

Our study started with an awareness of a major problem, namely, there is currently a lack of guidelines of how software can be used to support (collaborative) reflection during design thinking projects. This lack is problematic because, in creative work, individuals, as well as entire teams, need to continuously reflect on their actions and outcomes. For suggesting a solution, we conceptualize two main sources: First, we aim to theoretically ground our prototype by making use of the reflection theory, which tries to understand how people can learn from experiences and derive plans for the future. Second, we aim to gather experiences, challenges, and recommendations in terms of creative projects. Therefore, we selected three persons who already carried out at least one design thinking project in which they learned the design thinking method (from a professional coach) and worked on a concrete problem. Each of the three interviews contained five

parts: (1) reflection of the entire project, (2) communication during the project, (3) challenges during the project, (4) logging of activities, and (5) time and coordination issues during the project.

Afterward, in the development phase, we draw on the findings from the conceptualization and started to implement our software prototype that translates the gathered requirements into a concrete IT-artefact. To evaluate the prototype already in an early stage [36], we carried out a formative evaluation [37]. Therefore, we conducted a workshop with a group of five master-level Information Systems students who completed a design thinking project that deals with innovating marketing concepts of a small city. As we focused on evaluating the features in particular, we created a short guide comprising explanations of the key features (e.g., ‘private messaging’—‘the tool allows sending private messages to team members’ and ‘the tool allows reading received messages directly in the user’s dashboard’). As a result of this, we learned that statistics regarding the usage, task prioritization, and email notifications should be provided.

3.2. Design cycle 2 (refinement)

Based on the lessons learned from design cycle 1, we revised the conceptualization and refined the software prototype. To evaluate the results, a workshop was performed in which the revised key features are discussed with another group of four master-level Information Systems students. These students also carried out a design thinking project in which they had to innovate a B2B online platform from an industry partner. Similar to the evaluation in the first cycle, our evaluation focuses on verifying the applicability and completeness of the refined features. Therefore, we revised our guide comprising the descriptions of the key features and asked the workshop participants whether they think that features are missing, need to be revised, or are especially useful. As a result, we could identify only limited potential for further refinement which deals, for example, with (a) archival storage (i.e., history of prior project results), (b) group-based reflection (i.e., discussion about certain phases), and (c) user forum (i.e., communication within a group through web-based forums).

3.3. Design cycle 3 (current state)

Finally, we again analyzed the lessons learned and consolidated the entire knowledge gathered during the prior cycles to suggest a revised conceptualization, which is applied to our web-based software prototype. For evaluation, we provided our prototype to two

groups of master-level Information Systems students who had to accomplish a project which took about six months: The first group of six participants was concerned with creating an innovative event format for an industry partner offering car tires. The second group of five participants had to create an innovative approach that allows communicating information across several channels from a local organization that deals with the promotion of regional economic development. Thus, we demonstrate the applicability of our prototype with eleven participants [38].

4. Deriving design requirements

In order to provide justifications for the design, theoretical work from the natural or social science should be employed to derive underlying requirements [39]. Therefore, our design builds on two streams, namely (1) the basic concepts from reflection theory as well as (2) expert interviews and related literature. We specify an expert as someone who has already finished at least one design thinking project within a natural environment like in collaboration with an industry partner. Next, we describe both sources in more detail.

To offer a common basis to all participants involved in a reflection process, information and experiences need to be visible [18]. Generally, the act of designing consists of naming and framing (e.g., make visible what you see), moving (e.g., change the design), and evaluating (e.g., judge about moves) [27]. In problematic and complex situations, naming and framing objects of interest is an important task to make them manageable [12], [27]. Based on the formulation of information and the problem space, designers are able to find new solutions [14]. Here, software is assumed as worthwhile because it can store and show relevant information during the reflection process [15]

‘Creative design’ presumes continuous developing and refining both the problem specification and the ideas for a solution [43]. Thus, also design thinking projects are iteratively (e.g., [10], [24], [44]), and each project group creates a unique design path [45]. To contribute to the comprehensibility of iterations and paths, all of the interviewed experts argued that a graphical visualization of the progress is essential because, for example, “at the beginning of each session, we collaboratively discussed the current state and the past path of our process based on a visualization of the applied procedure model.” This kind of graphical representation is important from an individual view (i.e., reflecting on what an individual has done so far including, for example, which methods have been used) as well as from a collective view (i.e., reflecting on what a group has done so far).

In addition to the visualization of the project path, software should enable the entire progress including aspects such as single meeting protocols, selected methods, and discussions to be reflected. Therefore, project statistics are needed to visualize how many methods have been used, how many tasks have been completed, and how many tasks still have to be done. In order to contribute to a consistent way of visualizing relevant information that arises during design thinking projects (single sessions and meetings), a standardized template for reflecting the entire project as well as for single meetings, etc. [45] has to be provided. This is evident by statements like “we actually did not know which information might be relevant for future steps”. Accordingly, *a (standardized) visualization of project-related information including the progress of an entire design thinking project, the sessions held, and the methods applied is required.*

Previous research has determined that asking a group of reflection participants questions facilitates the process of collaborative reflection [18]. Furthermore, it provides guidance on the actual process of reflection [17]. Referring to this study’s context, as a first step, to adequately guide the progress of a design thinking project, an underlying procedure model describing of the main phases to be carried out has to be selected. Especially to achieve a goal within a team, a collaborative process must be exactly specified (e.g., [4], [42]). Although there are many similarities in the proposed procedure models every model has its own specifics [10], [11], which need to be respected during the project, and thus, users should be able to select the most suitable for their purpose. During the interviews, experts argued that “the beginning of the project was very hard because we often had to look up details regarding the process itself”, and “(...) sometimes we didn’t really know what exactly to do next.” As manifold methods and techniques can be applied in design thinking [4], users are sometimes overwhelmed by the task of selecting one in a certain phase of the underlying procedure model. Therefore, the experts ask for standardized templates to log information in terms of methods and techniques in a more unified manner. In addition to providing descriptions of those methods and techniques, experts ask for recommendations of methods and techniques, which are suitable for a certain design thinking phase or activity. Accordingly, *guidance through the abstract design thinking process as well as through specific phases and the application of methods and techniques is required.*

Another essential task is exploring consequences that might occur from a particular action. After identifying such consequences, they need to be assessed, for instance, what is bad or what is good. Schön [14, p. 6] argued that “in the absence of such

qualitative judgments (...) designing would have no thrust or direction [*and thus*] would be entirely unmotivated”. In general, it is assumed that through the act of designing objects implicit knowledge can be elicited. This, for instance, is evident by the concept of *knowing-in-action* which says that designers often make innumerable judgments of quality during the design process [12]. Referring to this study’s purpose, evaluating, for example, different ideas, prototypes or assumptions is a crucial task that also has impacts on the entire project. Thus, stakeholder should be enabled to continuously evaluate certain design activities and outcomes. To do so, it might be helpful to take different perspectives on a situation [41]. As people have to deal with plenty of different domains and qualities during the design process, they cannot consider all domains at the same time (e.g., limited information processing capacity). To overcome this, they often start by taking one perspective on a situation or an object and explore further relevant perspectives in consecutive steps [14]. As another example, Prilla [19, p. 3] argues that reflection needs “communication among reflection partners to [...] discuss perspectives, and agree on common solutions”. By taking perspectives, a designer might handle complex situations such as in creative projects more easily than by being confronted with the holistic problem at once.

While design thinking is a collaborative task [3], [25], allowing interaction and communication among the group members are crucial. Therefore, features for the exchange of messages between single users as well as the discussion of topics with all users of a certain team are required. If implemented, those features enable feedback to be provided and evaluation to be made. Moreover, an expert stated that “knowing each other within a group is very important, which can be for example supported by user profiles, etc.” Consequently, tools should implement a user and role management including the assignment of roles. Accordingly, *collaborative evaluation of the design thinking process and the design thinking outcomes by considering different perspectives on a situation and discussing their feedback is required.*

Exploring and imaging alternative solutions, as well as a range of appropriate concepts that can be compared to each other, is a crucial activity for critical reflection [31]. To create and compare such alternative concepts, designers usually make use of repertoires of prototypes and solutions from earlier experiences and situations [14]. Doing so, a designer selects one or more prototypes from his or her repertoire, compares it with the current situations and its restrictions, and adapts it to solve the problem at hand or to provide a new solution. For this, software can be used to extend the designer’s repertoire of concepts and prototypes [14].

Accordingly, *storage and usage of previous design thinking solutions and prototypes are required.*

Overall, an essential prerequisite for reflection is that people make experience explicit, share them, and discuss them to gain insights for future actions [19], which thus, should be considered across all features. In such situations, software support is assumed as valuable [14], because it enables remotely, cross-boundary, and time-independent discussion, a shared storage of information [17], and awareness by sending updates to the reflection team [19].

5. Instantiating the design requirements

5.1. Design features

In the following, to reflect whether and how our software prototype addresses the design requirements obtained above, we describe design features that are implemented through our prototype. For presentation purpose, these features are ordered according to the four design requirements (see Table 1).

A *(standardized) visualization of project-related information including the progress of an entire design thinking project, the sessions held, and the methods applied*: To provide novel facts (e.g., new tasks and messages) as well as to summarize the current progress of a project, a user dashboard is implemented. This dashboard comprises the graphical visualization of the progress (based on a selected underlying procedure model), a team calendar, an overview of current tasks and messages as well as project statistics (*DF1*). In order to enable the entire project progress and prior path including decisions and phases passed through to be traced, we use a so-called ‘interactive knowledge-map’. This map represents the selected procedure model and adds small flags on the representation to

graphically present in which phases sessions were performed (i.e., highlighting sessions protocols to the procedure). It can be selected by individuals (i.e., reflecting and tracing the personal path) as well as by groups (i.e., reflecting the group-based path) (*DF2*).

As information should be presented in a purposeful schema that helps participants to reflect on a specific situation or design thinking activity, we provide a session template that enables project-relevant information to be visualized (*DF3*). These templates contain relevant parts including general data such as participants, length, and location as well as task-related information (e.g., assignment of tasks and deadlines), and reflection (e.g., retro perspective summary and discussion of the important points). In addition to creating these protocols, a feature allows for automatic exporting the entire project information like into a PDF-file that reports an overview of the group, session protocols, and individual reflections.

Guidance through the abstract design thinking process as well as through specific phases and the application of methods and techniques: As stated before, different underlying procedure models for design thinking can be applied [10-11], [49]. To come up with a procedure that already comprises activities for preparing and managing such projects, we created a model which is inspired by well-accepted models from HPI [11] and Brown [1] (see Figure 2, left). Our model consists of six phases (i.e., problem research, problem definition, idea generation, idea elaboration, prototype implementation, and prototype evaluation) among three spaces (i.e., problem space, solution space, and implementation space). Each of the three spaces differentiates between a convergent and a divergent phase. Additionally, two phases for preparation (i.e., promote creative, out-of-the-box thinking) and project management (i.e., support communication and collaboration within a team) are added that seek to

Table 1. Matching of design requirements and design features

Design requirements	Design features implementing the requirements
A(standardized) visualization of project-related information including the progress of an entire design thinking project, the sessions held, and the methods applied.	<ul style="list-style-type: none"> ▪ User dashboard (project facts, progress, calendar, etc.) (<i>DF1</i>) ▪ Individual and group-based project path visualization (<i>DF2</i>) ▪ Visualization of session protocols (what needs to be reflected?) (<i>DF3</i>)
Guidance through the abstract design thinking process as well as through specific phases and the application of methods and techniques.	<ul style="list-style-type: none"> ▪ Visualization of design thinking procedure model (<i>DF4</i>) ▪ Method recommendations based on current phase (<i>DF5</i>) ▪ Method rating (i.e., reflection on the applicability) (<i>DF6</i>) ▪ Assignment of tasks to users (<i>DF7</i>)
Collaborative evaluation of the design thinking process and outcomes by considering different perspectives on a situation and discussing their feedback.	<ul style="list-style-type: none"> ▪ Project-based Wiki, discussion forum (team-based) (<i>DF8</i>) ▪ Chat (team-based), private messaging, and email notification (<i>DF9</i>) ▪ User profiles (including competencies and interests) (<i>DF10</i>)
Storage and usage of previous design thinking solutions and prototypes.	<ul style="list-style-type: none"> ▪ Examples of results and prototypes (<i>DF11</i>) ▪ Application cases of methods and techniques (<i>DF11</i>)

leverage continuous reflection in such projects (**DF4**).

For guiding how and when to apply a certain method or technique, we implemented a ‘method database’. In this database, we (drawing on an extensive literature search) assigned methods to our underlying six-phase procedure model and verified the assignment by interviewing a professional design thinking coach. Thus, users of the prototype are able to select a method from a set of possible ones based on the current phase (**DF5**)—in further steps, this can be enhanced by using automated recommender systems for example. Furthermore, each method is described in more detail, examples from prior projects are integrated, and a rating schema is provided so that the users can evaluate whether a method was appropriate for a certain phase/activity or not (**DF6**).

Because those activities need to be coordinated and future actions need to be specified (based on learnings derived from continuous reflection), we enable tasks to be created and to be assigned to specific users. These tasks can be added with information such as deadlines, descriptions, and current state of a task (**DF7**).

Collaborative evaluation of the design thinking process and the design thinking outcomes by considering different perspectives on a situation and discussing their feedback: Team members need to interact with each other to share views and judgments on certain situations, outcomes, or the design process. To elevate this collective reflection, we implemented different features: First, for exchanging and storing information within an entire design thinking team, a project wiki and a discussion forum (**DF8**) is provided. Second, for exchanging messages between selected team members, we implemented a chat as well as features for private messaging which automatically send email notifications (**DF9**).

Reflecting presumes taking different views to make complex problems more easy to handle. Hence, design thinking users need to select team members that have different backgrounds to allow perspectives to be considered. Our software prototype provides features for creating user profiles that help to specify own competencies, interests as well as experiences (e.g., work experiences). Thus, teams are enabled to form a proper team for solving a certain problem (**DF10**).

Storage and usage of previous design thinking solutions and prototypes: Allowing users to draw on previous results and prototypes, we initially provide project examples and use cases of different methods and techniques that can be applied (**DF11**). However, this is a preliminary step that can be extended by, for example, adding results from other projects, reference models (i.e., best practice solutions for certain problems), or business model patterns that help to specify the main constructs of a specific business.

5.2. Expository instantiation

In order to promote easy, cross-boundary, and location-independent access and collaboration, our prototype is implemented as a web application, which is based on an open source project [46]. As illustrated in Figure 2, our prototype implements a user dashboard that visualizes the entire project path in an interactive map (i.e., pins visualize a certain activity or document). In doing this, it allows users reflecting on single outcomes as well as the design process. Via different tabs that are placed in the top of the software, details on aspects such as methods and techniques, sessions protocols, and project management including task sharing and coordination can be opened.

6. Demonstration and evaluation

Design science evaluation can be conducted *ex-ante* or *ex-post* the design as well as in an *artificial* or *naturalistic* setting [37]. For evaluating the software prototype already in the early stages, we carried out two *ex-ante* evaluations in which a group of design thinking participants assessed the features implemented (see Section 3). For *ex-post* evaluation, we performed three main activities: First, we demonstrate the prototype within a use case. Second, to obtain information regarding the use of the software, we analyze log files and data from the software. Third, we discussed the prototype with an industry partner interested in software-supported design thinking.

First, we seek to demonstrate the applicability of the implemented prototype within a real-world situation [37], here with master-leveled students (mostly enrolled in Information Systems) as well as a trained design thinking coach. As our artifact is still in a prototype stage, the validation in an artificial setting is appropriate [38]. Following this, we conducted a design thinking project across six months in which the participants apply the prototype demonstrating that the artifact is useful for its intended purpose. Generally, we could observe that the software (and thus also the design features) supports continuous reflection on intermediate results and outcomes as well as the process. In particular, it has positive effects on communication and collaboration (e.g., parallelizing tasks within a project group and more focused discussions based on the progress visualization). Moreover, providing detailed descriptions of methods and techniques with concrete examples that illustrate how they can be employed, was helpful because users can make more informed decisions regarding what method they would like to use. In contrast to these positive effects, we could observe that some

participants often deal with handling the software in order to type in results, progress details, etc. (mostly in analog sessions in which the prototype was used). In doing this, he or she cannot contribute to the group.

Second, besides demonstrating the applicability of the prototype, we investigated, based on automatic-generated log files and data from the software, how often the features have been used: As one example out of three exemplary groups (each between from four to six participants), one group specified a total of 38 tasks across the six participants, used a total of 16 different creativity techniques, and created 16 sessions protocols. Overall, across the three groups, 51 sessions protocols were published, 59 techniques and methods were employed (24 divergent, 20 convergent, 9 project preparation, and 6 project management). As lessons learned for future research, although our prototype enables various ways to communicate and share information, the participants stated that most communication was carried out with message services on their smartphones. Accordingly, the integration of further applications (like those that are used by the participants anyway) in our prototype or how these can be used, requires extensive evaluation in future steps.

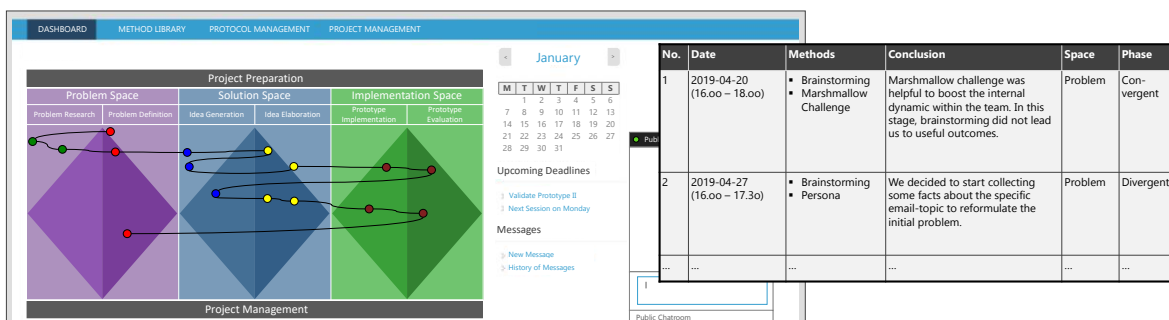
Third, for further practical relevance, we held a workshop in which we discuss our prototype with two CEOs of an industry partner that focuses on generating ideas for innovative start-ups. In a nutshell, as a key finding, the CEOs stated that the features implemented are useful for practical applications, and thus, aim to implement this study’s design knowledge into concrete IT-solutions that they offer to customers.

7. Conclusion

The present study seeks to build a software artifact that allows users to collaboratively reflect on and in design thinking projects. Therefore, we draw on both theoretical and empirical grounding. We make three main contributions: First, we derive design requirements that provide knowledge for the class of

creativity tools which can be instantiated by further artifacts. Second, we propose design features that indicate how the design requirements can be instantiated. Even though this study focused on software, reflection is important in any problem-solving situation, and thus, can be transferred to analog tools as well, for instance, to support business model development sessions (e.g., by employing the Business Model Canvas, [47]). Third, we present a software prototype that can be employed to assist design thinking projects as well as represent a source of knowledge for (re-)designing software.

This study is not free of limitations which opens plenty of future research avenues. First, the evaluations were mainly carried out in an educational setting. Although the entire project was attended by a professional coach and we discussed features with industry partners, further research should focus on the evaluation in a naturalistic setting (e.g., case study or expert panels). In this regard, additional experts should be taken into account to, for instance, validate or extend the current set of requirements—we initially specify an expert rather broad to draw on experiences from a larger group of participants. Second, it would seem worthwhile to perform an experiment that helps to understand and reveal which effects occur when applying a specific tool which seeks to aid reflection versus a general tool which is, for example, selected by the participants themselves. Third, the way in which a requirement is implemented is based on own decisions and choices. However, to face this, we have started to evaluate the prototype already in early stages, seek to contribute to the tractability of the implementation (e.g., by specifying design features), and conducted expert interviews. Fourth, to enable reflection in such tools, the participation of users is premised, and thus, users should provide comments and attend discussions to give feedback on certain elements. Hence, the system presumes users who are contributors. Fifth, recent studies have shown the importance of digitalization in creative industries [48] as well as the



Note: dashboard—left, session reflection—right.

Figure 2. Illustration of software prototype

importance of ICT-enabled design thinking for digital innovations like by exploring ‘innovation affordance’ [50]. Following this idea, our prototype can be used to further investigate how software-supported design thinking can be made more useful to the individual and context-specific project team’s demands.

Overall, our findings provide first insights about how visual tools might promote reflection of creative projects, and thus, opens avenues for future research such as investigating in which settings software might be more beneficial than paper-&-pen versions or how to best possible involve each team member in the reflection processes without being hindered by operating a tool. In particular, we plan to integrate results (e.g., design knowledge) from other studies concerning software-supported reflection (e.g., [51]).

Acknowledgments: This research was conducted in the scope of the research project “SmartHybrid—Process Engineering” (ZW 6-85003451) which is funded by the European Regional Development Fund (ERDF), the State of Lower Saxony, and the NBank. We would like to thank them for their support.

8. References

- [1] T. Brown, “Design Thinking“, Harvard Business Review, 2008.
- [2] F. Müller-Wienbergen, O. Müller, S. Seidel, and J. Becker, “Leaving the Beaten Tracks in Creative Work – A Design Theory for Systems that Support Convergent and Divergent Thinking“, Journal of the Association for Information Systems, Vol. 12, No. 11, 2011, pp. 714–740.
- [3] T. Potthoff et al., “Collaborative Service Blueprinting for Design Thinking: Evaluation of a Digital Prototype“, in Proceedings of Hawaii International Conference on System Sciences, 2018.
- [4] C. Lattemann, D. Siemon, D. Dorawa, and B. Redlich, “Digitization of the Design Thinking Process Solving Problems with Geographically Dispersed Teams“, in Design, User Experience, and Usability: Theory, Methodology, and Management, 2017, pp. 71–88.
- [5] F. Engberts and H. Borgman, “Application of Design Thinking for Service Innovation: Current Practices, Expectations and Adoption Barriers“, in Proceedings of the Hawaii International Conference on System Sciences, 2018.
- [6] W. Brenner, F. Uebornickel, and T. Abrell, „Design thinking as mindset, process, and toolbox“, in Design Thinking for Innovation, Springer, 2016, pp. 3–21.
- [7] D. Dunne and R. Martin, “Design thinking and how it will change management education: An interview and discussion“, Academy of Management Learning & Education, Vol. 5, No. 4, 2006, pp. 512–523.
- [8] R. Buchanan, “Wicked Problems in Design Thinking“, Design Issues, Vol. 8, No. 2, 1992, pp. 5–21.
- [9] Design Council, “The Design Process: What is the Double Diamond?“. Available at: www.designcouncil.org.uk [Accessed: 23-May-2019].
- [10] d.School, An Introduction to Design Thinking. 2018.
- [11] HPI, Design Thinking-Prozess. 2018.
- [12] D.A. Schön, The reflective practitioner: How professionals think in action. London, UK: Basic Books, 1983.
- [13] D.A. Schön and V. DeSanctis, The reflective practitioner: How professionals think in action. Taylor & Francis, 1986.
- [14] D.A. Schön, “Designing as reflective conversation with the materials of a design situation“, Knowledge-based Systems, Vol. 5, No. 1, 1992, pp. 3–14.
- [15] B. Renner et al., “Computer-supported reflective learning: how apps can foster reflection at work“, Behaviour & Information Technology, 2019, pp. 1–2.
- [16] M.K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, “Action Design Research“, MIS Quarterly, Vol. 35, No. 1, 2011, pp. 37–56.
- [17] B. Renner, M. Prilla, U. Cress, and J. Kimmerle, “Effects of Prompting in Reflective Learning Tools: Findings from Experimental Field, Lab, and Online Studies“, Front. Psychology, Vol. 7, 2016.
- [18] M.W. Daudelin, “Learning from experience through reflection“, Organizational Dynamics, Vol. 24, No. 3, 1996, pp. 36–48.
- [19] M. Prilla, “Supporting collaborative reflection at work: A socio-technical analysis“, AIS Transactions on Human-Computer Interaction, Vol. 7, No. 1, 2015, pp. 1–17.
- [20] B. Kuechler and V. Vaishnavi, “On theory development in design science research: Anatomy of a research project“, European Journal of Information Systems, Vol. 17, No. 5, 2008, pp. 489–504.
- [21] N. Cross, Design thinking: Understanding how designers think and work. Berg, 2011.
- [22] P.G. Rowe, Design thinking. MIT Press, 1991.
- [23] T. Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. Harper Collins, 2009.
- [24] T. Brown and B. Katz, “Change by Design“, Journal of Product Innovation Management, Vol. 28, No. 3, 2011, pp. 381–383.
- [25] B. Redlich, D. Dorawa, D. Siemon, and C. Lattemann, “Towards Semi-Virtual Design Thinking-Creativity in

- Dispersed Multicultural and Multidisciplinary Innovation Project Teams”, in Proceedings of the Hawaii International Conference on System Sciences, 2018.
- [26] M. Lynch and M. Metcalfe, “Reflection, pragmatism, concepts and intuition”, *Journal of Information Technology Theory and Application*, Vol. 7, No. 4, 2006.
- [27] W. Visser, “Schön: Design as a reflective practice”, *Collection*, No. 2, 2010, pp. 21–25.
- [28] J. Dewey, *How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process*. Boston, USA: D.C. Heath and Company, 1933.
- [29] D. Boud, R. Keogh, and D. Walker, “Promoting reflection in learning: A model”, *Reflection: Turning experience into learning*, 1985, pp. 18–40.
- [30] K. Grushka, J. H. McLeod, and R. Reynolds, “Reflecting upon reflection: Theory and practice in one Australian university teacher education program”, *Reflective Practice*, Vol. 6, No. 2, 2005, pp. 239–246.
- [31] M. van Woerkom and M. Croon, “Operationalising critically reflective work behaviour”, *Personnel Review*, Vol. 37, No. 3, 2008, pp. 317–331.
- [32] J. Iivari, “Reflections on Reflective Systems Development”, *Scandinavian journal of information systems*, Vol. 10, No. 1, 1998.
- [33] N. Hatton and D. Smith, “Reflection in teacher education: Towards definition and implementation”, *Teaching and teacher education*, Vol. 11, 1995, pp. 33–49.
- [34] M. Prilla, “Collaborative reflection support at work: A socio-technical design task”, in Proceedings of the European Conference on Information Systems, Tel Aviv, Israel, 2014.
- [35] M. Prilla and B. Renner, “Supporting collaborative reflection at work: A comparative case analysis”, in Proceedings of the international conference on supporting group work, 2014, pp. 182–193.
- [36] C. Sonnenberg and J. vom Brocke, “Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research”, in *Design Science Research in Information Systems*, Vol. 7286, Berlin, Heidelberg: Springer, 2012, pp. 381–397.
- [37] J. Venable, J. Pries-Heje, and R. Baskerville, “FEDS: a Framework for Evaluation in Design Science Research”, *European Journal of Information Systems*, Vol. 25, No. 1, 2016, pp. 77–89.
- [38] K. Peffers, M. Rothenberger, T. Tuunanen, and R. Vaezi, “Design Science Research Evaluation”, in *Design Science Research in Information Systems*, Vol. 7286, Berlin, Heidelberg: Springer, 2012, pp. 398–410.
- [39] J.G. Walls, G.R. Widmeyer, and O.A. El Sawy, “Building an Information System Design Theory for Vigilant EIS”, *Information Systems Research*, Vol. 3, 1992, pp. 36–59.
- [40] M. Prilla, A. Nolte, O. Blunk, D. Liedtke, and B. Renner, “Analyzing collaborative reflection support: a content analysis approach”, in Proceedings of the European Conference on Computer Supported Cooperative Work, Oslo, Norway, 2015, pp. 123–142.
- [41] D. List, “Reflection on the Future: Its Possibility and Usefulness”, *Journal of Information Technology Theory and Application*, Vol. 7, No. 4, 2006.
- [42] G.J. de Vreede and R.O. Briggs, “Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks”, in Proceedings of the Hawaii International Conference on System Sciences, 2005.
- [43] K. Dorst and N. Cross, “Creativity in the design process: co-evolution of problem–solution”, *Design Studies*, Vol. 22, No. 5, 2001, pp. 425–437.
- [44] T. Brown, “Design Thinking”, *Harvard business review*, Vol. 86, 2008, pp. 84–92.
- [45] A. Menning, T. Beyhl, H. Giese, U. Weinberg, and C. Nicolai, “Introducing the LogCal: template-based documentation support for educational design thinking projects”, in International Conference on Engineering and Product Design Education, Twente, Netherlands, 2014.
- [46] J. Hofer, T. Schoormann, J. Kortum, and R. Knackstedt, “Ich weiß was ihr letzte Sitzung getan habt - Entwicklung und Anwendung eines Softwarewerkzeuges zur Dokumentation von Design Thinking-Projekten”, *HMD Praxis der Wirtschaftsinformatik*, Vol. 56, 2019, pp. 160-171.
- [47] A. Osterwalder and Y. Pigneur, *Business Model Generation: A Handbook for visionaries, game changes, and challenges*, John Wiley & Sons, 2010.
- [48] K. Kutzner, T. Schoormann, and R. Knackstedt, “Digital Transformation in Information Systems Research: A Taxonomy-based Approach to Structure the Field”, in Proceedings of the European Conference on Information Systems, Portsmouth, UK, 2019.
- [49] T.B. Lembcke, A.B. Brendel, and L.M. Kolbe, “Make Design Thinking Teams Work: Einblicke in die Herausforderungen von innovativen Team-Kollaborationen”, *HMD Praxis der Wirtschaftsinformatik*, Vol. 56, 2019, pp. 135-146.
- [50] A. Shirish, I. Boughzala, and S.C. Srivastava, “Does technology-enabled design-thinking influence digital innovation? An innovation affordance perspective”, in Proceedings of the Hawaii International Conference on Systems Sciences, Hawaii, USA, 2019.
- [51] T. Schoormann, D. Behrens, and R. Knackstedt, “The Noblest Way to Learn Wisdom is by Reflection: Designing Software Tools for Reflecting Sustainability in Business Models”, in Proceedings of the International Conference on Information Systems, San Francisco, USA, 2019.