Designing Platforms for Crowd-based Software Prototype Validation: A Design Science Study

First Author
Affiliation
email@domain

Second Author
Affiliation
email@domain

Third Author
Affiliation
email@domain

Abstract

Designing a software product based on early user feedback aligns it faster with the user's needs rather than validating them after the development. This feedback can be provided iteratively on software prototypes before the development to judge the idea behind the product and save development resources. Here, crowdsourcing techniques can be used to collect feedback from many potential users. However, less research focused on software support for this crowd-validation process. Therefore, we conducted a design science research study with three design cycles to develop a platform for software developers to support the prototype validation process using the crowd. We present abstracted design knowledge in the form of design principles and an overall solution concept together with a situated implementation of design features and a software artifact. Our research contributes knowledge to software designers in research and practice designing new and extending existing tools with iterative crowd-validation support.

1. Introduction

Nowadays, the constantly chaining VUCA (Volatility, Uncertainty, Complexity, Ambiguity) world has a high impact on developing new software products. Due to the increasing market uncertainties [1] and the wish for users to get integrated solutions for their problems instead of rare software products [2], early feedback from potential users in the market is essential for building successful software products. This feedback can be used to validate the most important assumptions about the software product before the actual development [3]. This validation can be used to decide to go or go not with an overall idea and guide the further development of certain software features. This is especially important in highly competitive markets like mobile ecosystems with millions of already developed software products (i.e., apps) [4].

This validation of the product, in turn, is possible with product discovery that aims to "quickly separate the good ideas from the bad to answer the question of which products, features or services should be developed to fulfill the needs of the customer" [5]. This discovery process, in turn, could be guided by management tools like Product Board[1]. Within the discovery, the current ideas about the product can be visualized using UI prototypes and evaluated by potential users. Those visualizations can be supported by prototyping tools like Figma[2]. However, a key challenge of effective product discovery is the access to the potential users during the discovery [5].

The access to users can be supported by using crowdsourcing. Crowdsourcing proposes the outsourcing of different activities to a large undefined set of users by using an open call [6]. To support such open calls, digital platforms for organizing the crowdsourcing and providing access to the crowd worker like Amazon Mechanical Turk Platform[3] have been established. Here, the crowd could also be used to support the different stages of the product development [7] like the product discovery. However, to the best of our knowledge, there is a gap in how to design platforms that support the prototype development based on the iterative feedback from a crowd of potential users and reduce uncertainties in product discovery. Therefore, we aim to gain knowledge about designing such a platform for crowd-validation by answering the following research question (RQ): *How to design platforms that integrate crowdsourcing techniques in the iterative validation of prototypes?*

To answer the research question, we conducted a design science research (DSR) study [8] with three design cycles to develop abstracted design knowledge and a situated implementation. For the abstracted design knowledge, we have developed nine design principles and an overall solution design. In this

[1] Product Board Tool: `https://www.productboard.com/`
[2] Figma Tool: `https://www.figma.com/`
[3] Amazon Mechanical Turk Platform: `https://www.mturk.com/`

design, the software developers iteratively validate their prototypes with the potential users (i.e., the crowd). For that, they receive the individual answers to predefined questionnaires together with support for aggregation and visualization. Moreover, incentive processes for motivating the users and approval processes for supporting the secrecy are included. For the situated implementation, we have developed concrete design features and an instantiation of a platform prototype. We evaluated both with an expert workshop and two user studies. With both contributions, we support software designers in research and practice to extend their software with crowd validation.

In the following, we first show the research background of our approach in terms of crowdsourcing and design principles (Sect. 2). Based on that, we introduce our research approach with the methodology and the conducted process (Sect. 3). Next, we show our abstracted design knowledge with the design principles and the overall solution design (Sect. 4). Based on that, we derive a situated implementation with the design features and the platform prototype (Sect. 5). Moreover, we show the user evaluation with the setting, the conduction, the interpretation, and current threats to validity (Sect. 6). Finally, we draw a conclusion and provide future work (Sect. 7)

2. Research Background

To build the foundation of our approach, we have provided a research background. For that, we present the usage of crowdsourcing in software development and explain design principles for digital platforms.

2.1. Crowdsourcing of Software Products

The development of software products is a resource-intensive task that can be improved with the iterative feedback of potential customers [9]. To gather that feedback, one concept is crowdsourcing which describes the outsourcing of value-creating activities from a company to a large undefined set of users by using an open call [6]. Crowdsourcing has been established in different research directions like crowd testing, crowd funding, crowd ideation, crowd logistic, crowd production, crowd promotion, and crowd support over the last years [10]. Here, the sub directions of crowd tests and ideation are mostly related to the solution for our approach.

Crowd Testing can be used to evaluate different running software products with the users. Here, CrowdStudy [11] is an approach to allow developers to test the usability of their web interfaces with the crowd workers of Amazon Mechanical Turk. Mechanical Turk is also used by CrowdCrit [12] to support designers to validate created posters in the form of uploaded images.

Crowd Ideation can be used to generate new and improve existing ideas for software products with the users. Here, a recent study by Shixuan et al. [13] analyses the cognitive load during the idea generation and convergence with the crowd under the manipulation of the task complexity, the idea representation, and the procedural guidance. Another study by Zaggl et al. [14] focuses on integrative solutions by reusing the already existing public knowledge of the crowd. ERICA [15] is a tool to use expert knowledge to validate diverse crowd answers. However, none of the approaches directly focused on the application area of prototypes.

2.2. Design Principles for Digital Platforms

In order to develop abstracted design knowledge, design science research (DSR) [16] has been established as one often used method. With DSR, a class of problems is solved by focusing on a specific problem and abstracting the results of the solution. To make those abstracted knowledge transferable to different problems, design principles (DP) can be used. Here, DPs capture and codify that knowledge in an explicit way by focussing on the implementer, the aim, the user, the context, the mechanism, the enactors, and the rationale [17]. DPs can be designed in a supportive way based on identified knowledge sources and the derivation of design requirements at the beginning or in a reflective way by extracting them directly from an instantiated software artifact [18]. Moreover, DPs can be formalized in different abstraction levels, directly impacting the researchers' reusability and practitioners' applicability [19]. In recent years, DPs for different software tools and crowd interactions have been proposed.

For **Software Tools**, those DPS describe the design knowledge from which features can be derived. Here, the Crowd-based Business Model Validation System [20] provides DPs to validate uncertainties in the business model development using crowdsourcing. Based on that, the Hybrid Intelligence Decision Support System for Business Model Validation [21] combines crowdsourcing with machine learning aspects to improve the validation. Moreover, a recent study by Schoormann et al. [22] works on DPs for tools to reflect sustainability in design thinking projects, where prototyping plays a major role. Last, a study by Reibenspiess et al. [23] designs DPs for an intrapreneurial platform to generate ideas.

For **Crowd Interactions**, those DPs describe the design knowledge from which interactions can be derived. Here, an approach by Tavanapour et al. [24]

provides DPs for crowd collaboration based on different intrinsic and extrinsic incentives. Moreover, a study by Chasin et al. [25] builds DPs for managing digital community currencies on software platforms. However, none of those developed DPs of the approaches directly focused on the application area of prototypes.

3. Research Approach

To answer our research question, we use design science research (DSR). For that, we explain the underlying methodology and show our applied process.

3.1. Design Science Research Methodology

For our research, we use DSR as it aims to solve a class of problems by developing a solution to a specific problem and then generalize that gained knowledge [26] based on the development and evaluation of a corresponding software artifact [16]. Here, we aim to solve the problem of crowd-validation of mobile application prototypes but also ensure that they can be generalized to related application areas. For this, we use design principles (DP) to codify the knowledge in a transferable way [17]. Moreover, we base our DSR on the opportunity creation theory (OCT) [27] as kernel theory to stick in line with similar approaches like business model validation [20] or venture ideation [28] from digital entrepreneurship [29]. Here, OCT originally states that businesses (and their products) are co-created under high uncertainty [28]. Therefore (product) development is an entrepreneurial process where assumptions have to be validated directly with the customer using exploration and exploitation.

According to Gregor and Hevner [26], we position the contribution type of our design principles as operational principles that can be transferred to other domains together with our software prototype as situated implementation of the artifact. Moreover, we use an exaptation according to the knowledge contribution framework by a refinement of the existing concept of crowdsourcing.

3.2. Design Science Research Process

For DSR, we use the cycle of Kuechler and Vaishnavi [8]. The cycle, as shown in Fig. 1, consists of the following five iteratively conducted steps. First, we identify the *(1) Awareness of [the] Problem* based on a real-world problem and provide a *(2) Suggestion* of a possible solution. Next, we work on the *(3) Development* of the software artifact and conduct an *(4) Evaluation* of it. Based on the evaluation results, another iteration is conducted, and/or our research contributions as *(5) Conclusion* are provided.

In the **First Cycle**, we got aware of the problem by conducting a literature review and tool analysis in the application areas of lean development, UI prototyping, and crowdsourcing to derive initial design requirements (DR) for our approach. Based on mapping the theoretical and empirical DRs, we suggested our first design principles (DPs) together with a preliminary concept. Out of that DPs we developed the first design features (DFs) and instantiated them in a software prototype. Last, we evaluated them in an online expert workshop (n=6), where we explained the overall concept, showed the software platform, and asked for feedback. Subsequently, we gave the experts access to the platform. We sent out a questionnaire to rate the importance of the DPs and provide feedback on the overall idea, the proposed solution, current drawbacks of the platform, and additional feedback.

In the **Second Cycle**, we took the lessons learned from the expert workshop together with additional literature to revisit the underlying DRs. Based on that, we also revisited our DPs and the suggested concept. This lead also to a redevelopment of the DFs and a complete new instantiation of the software prototype. We evaluated the DPs and the prototype in a student seminar on the lean development of mobile applications. Here, the students (n=14) were divided into different groups (g=6) to develop an idea for an app within the seminar iteratively. Here, one student per group needed to upload their prototype with questions to the platform. Next, every student gave feedback on two predetermined prototypes by answering the questions that could be used to improve the prototypes. Last, the students evaluated the prototype of the platform on the platform itself, by rating the importance of the DPs together with feedback on the overall idea, the proposed solution, current drawbacks of the platform, and additional feedback.

In the **Third Cycle**, which results are also shown within this paper, we took the lessons learned from the user study to revisit our DRs. Out of that, we improved the DPs and the overall solution concept. Moreover, we improved the DFs and the existing software platform based on those changes. We evaluated the DPs and the prototype similar to the second cycle in a student lecture for the systematic development of AR/VR applications. Within the lecture, the students (n=26) had a mini project where they needed to develop such an AR/VR application in a group (g=8). Again, one student needed to upload the prototype, each student needed to evaluate two predefined prototypes, and all students needed to evaluate the DPs together with the platform.

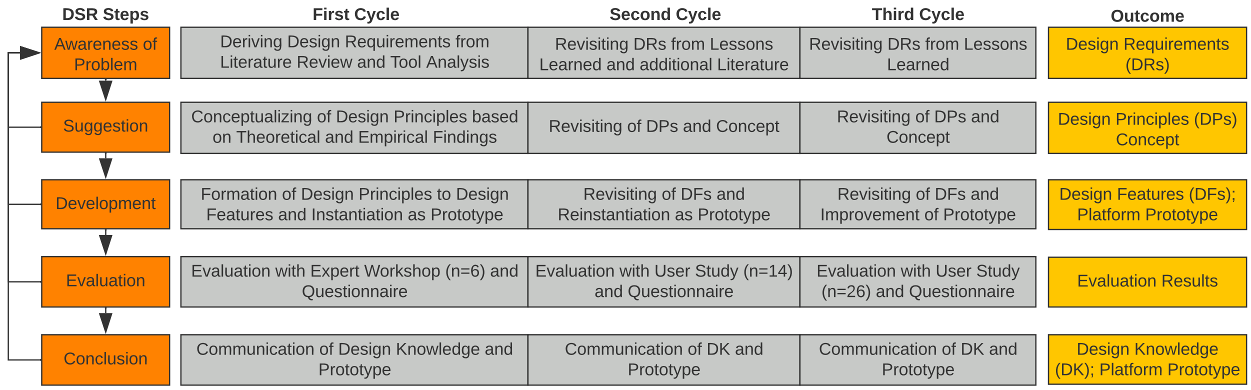


Figure 1. Research Approach (based on Kuechler and Vaishnavi [8])

4. Abstracted Design Knowledge

To make our study results transferable, we have developed abstracted design knowledge. For that, we show our revisited design principles and apply them to an overall solution design.

4.1. Design Principles (DPs)

We codify our knowledge during the design study within the DPs. Here, each DP shows a certain aspect of the platform and is based on the revisited DRs during the three cycles. Here, those DRs were derived from a literature review and tool analysis on the topics of lean development, UI prototyping, and crowdsourcing [blinded]. In the following, we show the nine DPs together with references to literature and tools that build the foundation for the mapped DRs.

DP1: User Variety states that *the solution should provide functions for integrating different internal and external users (e.g., platform user, crowd worker) to allow developers to participate with a heterogeneous group of users within the validation process*. Literature reasoned this because developers in the early product development have high uncertainties that can be validated by testing the underlying assumptions [30]. By using the knowledge of a crowd, the assumptions can be proofed [31], and the biases of the developers can be reduced [32]. Here, the users can come from internal sources like employees or external sources like Amazon Mechanical Turk[4].

DP2: Task Iteration states that *the solution should provide functions for conducting tasks iteratively to allow developers an incremental improvement of the prototypes over time*. Literature reasoned that user feedback could support the adjustment of product features [9] and the business model [1] to the market. For that, that feedback can be provided by a crowd of users like with ClickWorker [5] where the rapidness of the given feedback is a critical factor of success [30].

DP3: Prototype Diversity states that *the solution should provide functions for integrating different types of prototyping (e.g., mockups, click dummies) to allow developers a flexible choice for their current validation developments*. Literature reasoned that because depending on the stage of the product development, also different prototypes like textual descriptions, images, or click dummies can be used [33]. Here, different prototypes can ensure the refinement of the product features or business model over time [34]. For the visualizations, also external tools like Figma[6] can be used.

DP4: Feedback Diversity states that *the solution should provide functions for integrating different types of feedback (e.g., free texts, ratings) to allow developers a flexible choice for their current validation challenges*. Literature reasoned that depending on the type of the development stage also, different types of feedback are necessary [35]. Depending on the type of test, that feedback can consist of qualitative or quantitative information [34]. Different types of feedback are also integrated within the prototyping tool of UIGiants[7].

DP5: Filter Mechanisms states that *the solution should provide functions for the filtering between users and tasks to allow developers and users to shortlist evaluations based on specific criteria (e.g., skill set, interests)*. Literature reasoned that to ensure the quality of the feedback, the tasks must be just conducted by users of a relevant target group of the developer [36]. Conversely, users should see only tasks in which they

[4] Amazon Mechanical Turk: https://www.mturk.com

[5] ClickWorker: https://clickworker.com

[6] Figma: https://www.figma.com

[7] UIGiants: https://www.uigiants.com

are interested [37]. Amazon Mechanical Turk also uses two-sided filtering between the task provider and the crowd worker.

DP6: Aggregation Mechanisms states that *the solution should provide functions for aggregating and visualizing the feedback to allow developers to provide understandable and traceable improvements to the prototypes*. Literature reasoned that depending on the number of individual user feedback, it can be a time-consuming and challenging activity to process them. Here, the feedback should be provided to the developer in an aggregated form for fast processing [38]. Moreover, appropriate visualizations should support the developers in interpreting the feedback [39]. ClickWorker also aggregates the results of conducted tasks from different crowd workers.

DP7: Incentive Mechanisms states that *the solution should provide functions for supporting extrinsic and intrinsic incentives (e.g., rank lists, money) to allow developers to motivate users in the validation process*. Literature reasoned that giving valuable feedback is time-consuming and should ideally be done regularly [9]. Therefore, users should be offered extrinsic incentives like money or intrinsic incentives like fame [40]. While money is used as an extrinsic incentive by Amazon Mechanical Turk, intrinsic incentives like ratings and views are used by social media platforms like YouTube[8].

DP8: Non-Disclosure Mechanisms states that *the solution should provide functions for integrating non-disclosure agreements to allow developers to protect their prototypes from user thefts*. Literature reasoned that developing new ideas is a creative and challenging activity that often needs the collaboration of various stakeholders [41]. Depending on the trust between the developers and the users, non-disclosure agreements can be necessary for a more intensive idea exchange [42]. Those agreements are also often requested by clients on projects with a larger volume on the micro job platform Fiverr [9].

DP9: Governance Mechanisms states that *the solution should provide functions for integrating governance into the process to allow the platform owner to take necessary actions against developers and users that misusage the validation process*. Literature reasoned that providing valuable interactions between the developers and the users is the key task for the platform to stay successful. Good governance of those interactions will let the users stick much longer on the platform [43]. Here, governance in terms of policies, regulations, and accountability should be provided by the platform [44]. This, in turn, must be implemented in nearly every platform like Innocentive [10], which aims to solve problems.

4.2. Solution Design Concept

Out of the codified DPs, we conceptualize a solution design as shown in Fig. 2. It consists of the three roles of the *Developer*, the *User*, and the *Platform Owner* and the five components of the *Task Creation*, the *Task Conduction*, the *Task Evaluation*, the *Task Incentivisation*, and the *Task Approval*.

In the beginning, different *Developers* and *Users* (i.e., DP1) register to the platform, each with a specific profile. After that, the *Developer* creates a new or iterates an existing task (i.e., DP2) in the *Task Creation* by creating different types of prototypes (i.e., DP3), preparing different types of questions (i.e., DP4), and selecting specific criteria for users (i.e., DP5). Next, the *User* selects different tasks (i.e., DP5) in the *Task Evaluation*, depending on the approval process, executes the prototype (i.e., DP3), and provides feedback (i.e., DP4). This feedback is aggregated and visualized (i.e., DP6) in the *Task Evaluation* and displayed to the *Developer*. Based on that, the *Developer* can provide intrinsic and extrinsic incentives (i.e., DP7) to the *Users* in the *Task Incentivisation*. Moreover, the *Developer* can decide on an automatic selection of access to the prototypes with or without the usage of a non-disclosure agreement (i.e., DP8) in the *Task Approval*. Moreover, a manual selection is possible where the users ask for approval and get access to the prototype. Last, the *Platform Owner* governs the whole platform against misuse (i.e., DP9) by moderating developers/users and tasks.

5. Situated Implementation

To demonstrate our approach, we have developed a situated implementation. For that, we show our revisited design features and the implemented software platform.

5.1. Design Features (DFs)

We represent the features of our platform using DFs. Here each DP is translated to a set of DFs that can be directly implemented in the platform. In the following, we describe the DFs for each of the nine derived DPs.

For the **DP1: User Variety**, we allow the registration of developers and users both by a registration form (DF1) and a single sign-on service (DF2). Moreover, we provide user profiles with specific information like

[8] YouTube: `https://www.youtube.com/`
[9] Fiverr: `https://www.fiverr.com`
[10] Innocentive: `https://www.innocentive.com`

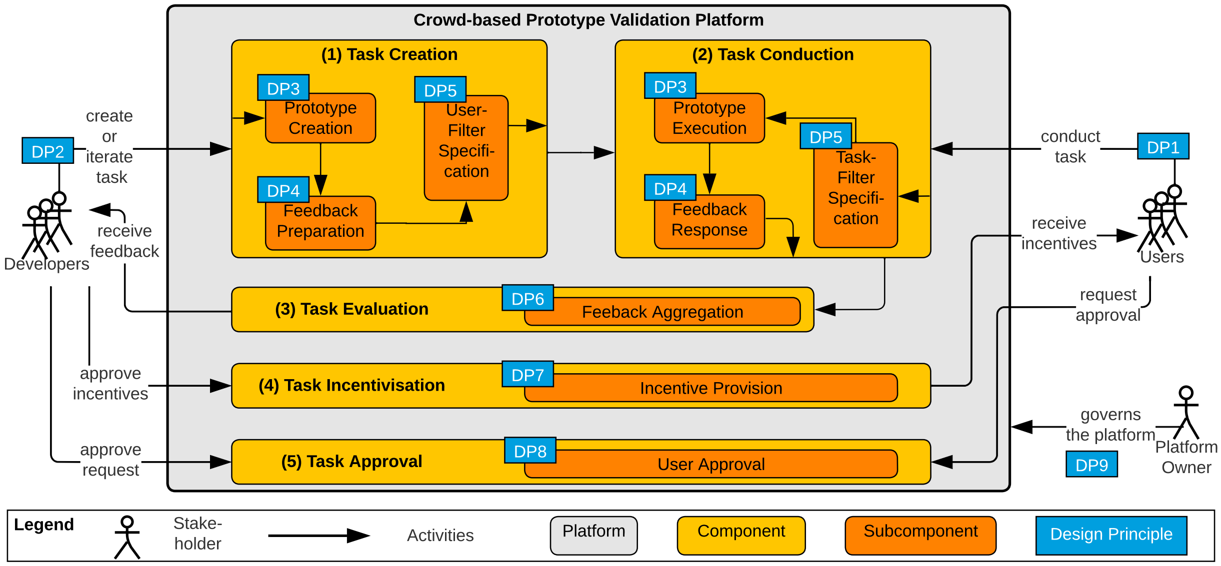


Figure 2. Solution Design Concept for a Crowd-based Prototype Validation Platform

skills (DF3) and a messaging system between different users (DF4). As **DP2: Task Iteration**, we provide the creation of tasks with essential information (DF5), the provision of feedback with a questionnaire (DF6), and the representation of feedback (DF7). For the **DP3: Prototype Diversity**, we allow the provision of prototypes as textual descriptions (DF8), uploaded images (DF9), and integration of external prototyping tools (DF10). In addition to that, the **DP4: Test Diversity** contains the test of single prototypes (DF11), the comparison of multiple prototypes (DF12), and the usage of split-tests (DF13). Here, also the questionnaire allows multiple types of questions like stars rating, thumbs-rating, radio buttons, and free text fields (DF14).

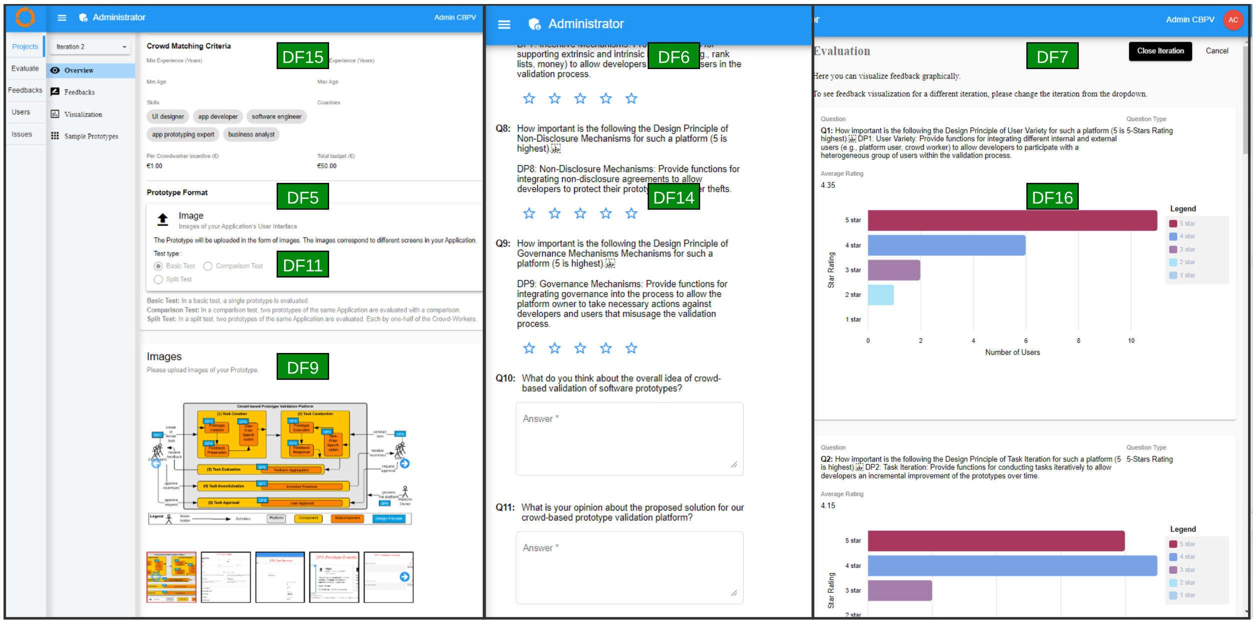
For the **DP5: Filter Mechanisms**, we support the adding of required user profile criteria to the tasks (DF15) and the shortlisting of tasks by the preferences of the users. As **DP6: Aggregation Mechanism**, we provide visualization charts of aggregatable answers to questions (DF16), investigation of individual feedback of each user (DF17), additional feedback for revealing unconsidered questions (DF18), and the comparison of split-test results (DF19). Based on that, the **DP7: Incentive Mechanism** allows the extrinsic motivation of sending virtual money (DF20) and the intrinsic motivation of publicly displaying the user's trustworthiness (DF21). As **DP8: Non-Disclosure Mechanisms**, we support the direct approval of tasks to all users (DF22), the deposit of a non-disclosure agreement that the users have to accept (DF23), and the manual approval of users (DF24). Finally, for the **DP9: Governance Mechanism**, we provide an admin control panel (DF25) together with the reporting of tasks and users (DF26).

5.2. Implemented Platform Prototype

Out of the developed DFs, we have implemented a software platform to test the features and the underlying principles with real users. While for the first cycle, we have developed a rapid prototype to test the overall idea, the second cycle was reimplemented in a scalable and extensible way so that improvements to the third cycle could be easily added.

For the implementation, our Crowd-based Prototype Validation (CBPV) Platform uses Angular[11] in the frontend, NestJS[12] in the backend, and PostgreSQL[13] as a database. Based on those core techniques, we build a microservice architecture so that new DPs and DFs can be easily implemented. Screenshots of our current version applied to the self-evaluation of the platform can be seen in Fig. 3. Here, the **a) Task Creation** shows the creation of tasks (i.e., DF5), the choice of a basic test for a single prototype (i.e., DF11), and the usage of uploaded images (i.e., DF9). The **b) Task Conduction** shows the provision of feedback (i.e., DF6) and the usage of star ratings (i.e., DF14). The **c) Task Evaluation** shows the representation of the feedback (i.e., DF7) and the aggregation of numerical feedback to questions (i.e., DF16).

[11]Angular: `https://www.angular.io`
[12]NestJS: `https://www.nestjs.com`
[13]PostgreSQL: `https://www.postgresql.org`



a) Task Creation   b) Task Conduction   c) Task Evaluation

Figure 3. Implemented Platform Prototype applied to the Self-Evaluation of the Concept

6. Evaluation Results

To evaluate our approach, we have conducted a user study as an evaluation for the third design cycle. For that, we explain the setting of our study, describe the derived results, analyze and interpret those results, and point out potential threats to validity.

6.1. Setting

We conducted our user study in a student lecture for developing AR/VR applications at [blinded] University. Here, the lecture aims to give students an overview of the different topics in the systematic development of AR/VR applications together with the skill-set to develop such an application from scratch. For that, those students have a mini-project where they grouped themselves into teams of 3-4 students to develop a prototype of their application.

At the beginning of the mini-project, the teams developed an idea (e.g., a JengaVR game, a smARt note app, an ARmomix cooking app) for the application. For that, they created a description, first screenshots, or a mockup of Figma together with a questionnaire of open questions for which they wanted to receive feedback. After that, one student from each team (g=8) uploaded their idea in the form of a prototype on the situated implementation of the CBPV Platform. Next, each student (n=26) provided feedback on the prototypes of two other groups by executing the prototypes and filling out the questionnaires. Here, the matching of the students to prototypes was made by us manually to provide a similar amount of feedback for every team. Last, the platform should be self-evaluated by every student (see Fig. 3 for an excerpt). Here, the prototype consists of images of the platform where the instantiations of the DPs on the platform are labeled. Moreover, the questionnaire provided a 5-stars rating question for each DP and four free text questions on the overall idea, the proposed solution, current drawbacks of the platform, and additional feedback.

6.2. Results

We present the user study results by referring to the created prototypes and filled-out questionnaire by most students (n=20) during the self-evaluation. For that, we divide between quantitative and qualitative results.

For the **Quantitative Results**, we have answers for the 5-stars rating questions for the nine DPs that mostly relate to the abstracted design knowledge of the crowd-based prototype validation. An overview of those results as boxplots is shown in Fig. 4. As an overall impression, we see that nearly every DP is rating as crucial for such a platform. The variety of users (i.e., DP1) and iteration of tasks (i.e., DP2) should be provided by every platform. Also, the diversity of prototypes (i.e., DP3) and feedback (i.e., DP4) together with the aggregation of feedback (i.e., DP6) that are specific for prototype crowd-validation

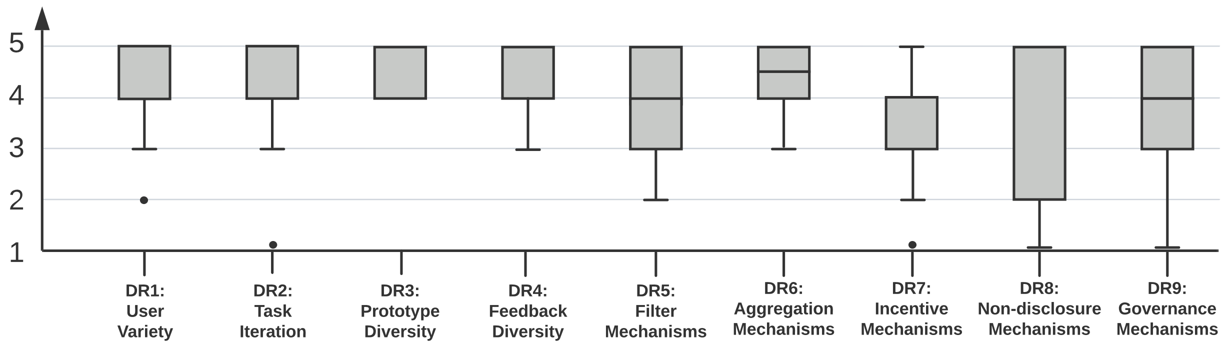


Figure 4. Evaluated Importance of Design Principles based on Boxplots

are rated as essential. The same holds for providing overall governance (i.e., DP9). The function for filtering (i.e., DP5) and incentives (i.e., DP7) are rated lower by the students as they got predetermined prototypes to validate and no additional incentives for the validations. Last, a higher discrepancy exists for the non-closure agreements (i.e., DP8), which some students could interpret as just additional overhead.

For the **Qualitative Results**, we have answers for the free text fields of the additional questions for the concept that are mostly related to the situated implementation of the platform prototype but can partially also be abstracted to the design knowledge. An overall impression of that feedback was that most of the students liked the overall idea of the platform. Just one student was curious if the additional effort would be worth the feedback, and one student commented that crowd validation should be just done in addition to regular customer interviews. Most of the feedback was regarding some general issues with the current version of the platform prototype, like better support for mobile web browsers, UI issues, simplified account management, or bug fixes. However, some feedback also suggested improvements to the most important features of the users, the prototypes, and the feedback. For the users, there was feedback to create groups for collaborative working on the prototypes and invite links to share the prototype with colleagues. For the prototypes, there was feedback to add additional types of non-visual prototypes and directly create clickable mockups on the platform. For the feedback, there was feedback for "if not, why" questions and blocks for Likert scale questions. Last, there was the wish to integrate the prototypes and the questions deeper.

6.3. Discussion and Implications

We interpret the user study results by analyzing the created prototypes by the teams and the filled-out questionnaire by the students. Out of that, we provide an analysis of the abstracted design knowledge and the situated implementation.

The **Abstracted Design Knowledge** refers to the developed design principles and the overall solution design. We currently see no major issues with the current set of principles. However, some DPs could be slightly improved in the future. For the variety of the users (i.e., DP1), the external users, and in the iteration of tasks (i.e., DP2), the incremental improvements could be described more precisely. Moreover, the deeper integration of prototypes (i.e., DP3) and feedback (i.e., DP4) could be mentioned. Next, the reasoning for the incentivization (i.e., DP7) and the non-disclosure agreements (i.e., DP8) could be improved. Last, based on the analysis of the created prototypes, a minor design principle that could be investigated in the future would be guidance in creating a task (e.g., choosing the best type of prototype, generating good questions for feedback).

The **Situated Implementation** refers to the design features and the implemented platform prototype. Here, we currently see a need to fix the current issues that were identified by the students. Moreover, we want to work on specific features that were mentioned during the evaluation. For that, in addition to single sign-on services (i.e., DF2) we want to allow the sending of invitation links for concrete task evaluations. Moreover, we want to allow the sharing of prototypes with other developers at the task creation (i.e., DF5). To improve the diversity of the prototypes, we want to add an internal prototyping tool in addition to the external one (i.e., DF10). Furthermore, the diversity of the feedback should be supported by multi-questions based on Lickert scales (i.e., DF14). As a larger project, we want to combine the creation of prototypes and the provision of questions deeper based on the integrated prototyping tool. Last, we want to implement the guidance in task creation as mentioned as a possible DP above.

6.4. Threats to Validity

We discuss our threats to validity according to Yin [45], who divides between Constructs Validity, Internal Validity, External Validity, and Reliability. **Construct Validity** refers to guaranteeing that the most verifiable case study results are based on the research question. To achieve that, we clarify the goal and purpose to the students of the lecture and provide additional explanations of the platform together with an email address for solving occurring problems. Nevertheless, there can be misunderstandings of the purpose, especially on the transferability of the DPs to different application domains. **Internal Validity** refers to establishing trustworthiness due to casual relationships during the case conduction. A threat here is the non-systematic literature review in the first cycle. While we cover different areas and use a technique like snowballing to reduce that threat, we can not completely ensure missing some literature. However, those issues should be reduced by conducting multiple design cycles. **External Validity** refers to the extent to which the results can be applied to other cases. A threat here is the evaluation in a student lecture and seminar because of the biased view of the students. Nevertheless, this should less affect the DPs due to the additional interviewing of experts. **Reliability** refers to the reproducibility of repeating the case study. For that, we record the whole expert workshop and export the raw data of all data created in the two user studies. While this increases the reliability of the study result, it could also harm the experts and students of providing negative feedback.

7. Conclusion & Future Work

The gathering of early feedback from users is vital to develop successful products. This feedback can be provided iteratively by a large number of crowd workers. However, currently it is unclear how to design platforms for such a crowd-validation of software products. Therefore, we have conducted a design science study with three design cycles to develop abstracted design knowledge (i.e., design principles, concept) and a situated implementation (i.e., design features, platform prototype) for a platform that solves that challenge. Our research contributes knowledge to software designers in research and practice designing new and extending existing tools with iterative crowd-validation support.

Our future work around the crowd-validation of software products is threefold: First, we want to improve the abstracted design knowledge and situated implementation based on interviews (e.g., interviewing experts from the industry). Second, we want to transfer the design knowledge to other application areas (e.g., validating prototypes in AR/VR). Third, we want to compare our solution against other existing approaches (e.g., combining a prototyping tool and questionnaire to manually collect feedback).

References

[1] R. G. McGrath, “Business Models: A Discovery Driven Approach,” *Long Range Planning*, no. 43, pp. 247–261, 2010.

[2] D. J. Teece, “Business Models, Business Strategy and Innovation,” *Long Range Planning*, vol. 43, no. 2-3, pp. 172–194, 2010.

[3] E. Lindgren and J. Münch, “Raising the odds of success: the current state of experimentation in product development,” *Inf. Softw. Technol.*, vol. 77, pp. 80–91, 2016.

[4] App Annie Inc, “The State of Mobile 2021.”

[5] J. Münch, S. Trieflinger, and B. Heisler, “Product Discovery – Building the Right Things: Insights from a Grey Literature Review,” in *Proceedings of ICE/ITMC*, pp. 1–8, IEEE, 2020.

[6] J. M. Leimeister, “Crowdsourcing,” *Controlling & Management*, vol. 56, no. 6, pp. 388–392, 2012.

[7] A. Tran, S. U. Hasan, and J.-Y. Park, “Crowd Participation Pattern in the Phases of a Product Development Process that Utilizes Crowdsourcing,” *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 266–275, 2012.

[8] 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, pp. 489–504, 2008.

[9] E. Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. USA: Crown Business, 2014.

[10] D. Durward, I. Blohm, and J. M. Leimeister, “Crowd Work,” *Business & Information Systems Engineering*, vol. 58, no. 4, pp. 281–286, 2016.

[11] M. Nebeling, M. Speicher, and M. C. Norrie, “CrowdStudy,” in *Proceedings EICS*, p. 255, ACM, 2013.

[12] K. Luther, J.-L. Tolentino, W. Wu, A. Pavel, B. P. Bailey, M. Agrawala, B. Hartmann, and S. P. Dow, “Structuring, Aggregating, and Evaluating Crowdsourced Design Critique,” in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 473–485, ACM, 2015.

[13] S. Fu, X. Cheng, T. de Vreede, G.-J. de Vreede, I. Seeber, R. Maier, and B. Weber, “Exploring Idea Convergence and Conceptual Combination in Open Innovative Crowdsourcing from a Cognitive Load Perspective,” in *Proccedings of HICCS 52*, AIS, 2019.

[14] M. A. Zaggl, Y. Sun, A. Majchrzak, and A. Malhotra, “Integrative Solutions in Online Crowdsourcing Innovation Challenges,” in *Proceedings of HICCS 54*, AIS, 2021.

[15] N. Quoc Viet Hung, D. Chi Thang, M. Weidlich, and K. Aberer, "ERICA," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1037–1038, ACM, 2015.

[16] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J Manag Inf Syst*, vol. 24, no. 3, pp. 45–77, 2007.

[17] S. Gregor, L. Kruse, and S. Seidel, "Research Perspectives: The Anatomy of a Design Principle," *Journal of the Association for Information Systems*, vol. 21, pp. 1622–1652, 2020.

[18] F. Möller, T. M. Guggenberger, and B. Otto, "Towards a Method for Design Principle Development in Information Systems," in *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry*, vol. 12388, pp. 208–220, Cham: Springer, 2020.

[19] H. Wache, F. Möller, T. Schoormann, G. Strobel, and D. Petrik, "Exploring the Abstraction Levels of Design Principles: The Case of Chatbots," in *Proccedings of WI 22*, AIS, 2022.

[20] D. Dellermann, N. Lipusch, and P. Ebel, "Developing Design Principles for a Crowd-Based Business Model Validation System," in *Designing the Digital Transformation*, vol. 10243, pp. 163–178, Springer, 2017.

[21] D. Dellermann, N. Lipusch, P. Ebel, and J. M. Leimeister, "Design principles for a hybrid intelligence decision support system for business model validation," *Electronic Markets*, vol. 29, no. 3, pp. 423–441, 2019.

[22] T. Schoormann, J. Hofer, and R. Knackstedt, "Software Tools for Supporting Reflection in Design Thinking Projects," in *Proceedings of the HICSS 2020*, AIS, 2020.

[23] V. Reibenspiess, K. Drechsler, A. Eckhardt, and H.-T. Wagner, "Tapping into the wealth of employees' ideas: Design principles for a digital intrapreneurship platform," *Information & Management*, p. 103287, 2020.

[24] N. Tavanapour and E. A. C. Bittner, "Towards Supportive Mechanisms for Crowd Collaboration – Design Guidelines for Platform Developers," in *Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration*, vol. 11591, pp. 353–372, Springer, 2019.

[25] F. Chasin, F. Schmolke, and J. Becker, "Design Principles for Digital Community Currencies," in *Proceedings of HICCS 53*, AIS, 2020.

[26] S. Gregor and A. R. Hevner, "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Quarterly*, vol. 37, no. 2, pp. 337–355, 2013.

[27] S. A. Alvarez, J. B. Barney, and P. Anderson, "Forming and Exploiting Opportunities: The Implications of Discovery and Creation Processes for Entrepreneurial and Organizational Research," *Organization Science*, vol. 24, no. 1, pp. 301–317, 2013.

[28] P. Vogel, "From Venture Idea to Venture Opportunity," *Entrepreneurship Theory and Practice*, vol. 41, no. 6, pp. 943–971, 2017.

[29] S. Nambisan, "Digital Entrepreneurship: Toward a Digital Technology Perspective of Entrepreneurship," *Entrepreneurship Theory and Practice*, vol. 41, no. 6, pp. 1029–1055, 2017.

[30] S. Blank, "Why the lean start-up changes everything," *Harvard Business Review*, no. 91, pp. 63–72, 2013.

[31] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, "Toward Data-Driven Requirements Engineering," *IEEE Software*, vol. 33, no. 1, pp. 48–54, 2016.

[32] K. Burmeister and C. Schade, "Are entrepreneurs' decisions more biased? An experimental investigation of the susceptibility to status quo bias," *Journal of Business Venturing*, vol. 22, no. 3, pp. 340–362, 2007.

[33] J. S. Linsey, E. F. Clauss, T. Kurtoglu, J. T. Murphy, K. L. Wood, and A. B. Markman, "An Experimental Study of Group Idea Generation Techniques: Understanding the Roles of Idea Representation and Viewing Methods," *Journal of Mechanical Design*, vol. 133, no. 3, 2011.

[34] H. H. Olsson and J. Bosch, "Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation," in *Software Business*, vol. 210, pp. 154–166, Springer, 2015.

[35] D. J. Bland and A. Osterwalder, *Testing business ideas*. Hoboken: John Wiley & Sons, 2020.

[36] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, "Developer Recommendation for Crowdsourced Software Development Tasks," in *2015 IEEE Symposium on Service-Oriented System Engineering*, pp. 347–356, IEEE, 2015.

[37] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with Mechanical Turk," in *Proceeding of CHI '08* (M. Czerwinski, A. Lund, and D. Tan, eds.), p. 453, ACM, 2008.

[38] D. Geiger, S. Seedorf, T. Schulze, R. C. Nickerson, and M. Schader, "Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes," in *AMCIS 2011 Proceedings - All Submissions*, AIS, 2011.

[39] A. Xu, S.-W. Huang, and B. Bailey, "Voyant: generating structured feedback on visual designs using a crowd of non-experts," in *ACM conference on Computer supported cooperative work & social computing*, pp. 1433–1444, ACM, 2014.

[40] L. Hammon and H. Hippner, "Crowdsourcing," *Business & Information Systems Engineering*, vol. 4, no. 3, pp. 163–166, 2012.

[41] P. Ebel, U. Bretschneider, and J. M. Leimeister, "Leveraging virtual business model innovation: a framework for designing business model development tools," *Information Systems Journal*, vol. 26, no. 5, pp. 519–550, 2016.

[42] A. O. Fanimokun, G. Castrogiovanni, and M. F. Peterson, "Developing High-Tech Ventures: Entrepreneurs, Advisors, and the Use of Non-Disclosure Agreements (NDAs)," *Journal of Small Business & Entrepreneurship*, vol. 25, no. 1, pp. 103–119, 2012.

[43] D. S. Evans and R. Schmalensee, *Matchmakers: The new economics of multisided platforms*. Boston: Harvard Business Review Press, 2016.

[44] G. Parker, M. van Alstyne, and S. P. Choudary, *Platform revolution: Platform Revolution: How Networked Markets Are Transforming the Economy - and How to Make Them Work for You*. New York: W.W. Norton & Company, 2016.

[45] R. K. Yin, *Case study research: Design and methods*, vol. 5 of *Applied social research methods series*. Los Angeles: Sage, 2009.