Scalable Design Evaluation for Everyone! Designing Configuration Systems for Crowd-Feedback Request Generation

Saskia Haug
saskia.haug@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Sophia Sommerrock
sophia.sommerrock@student.kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Ivo Benke
ivo.benke@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Alexander Maedche
alexander.maedche@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

ABSTRACT

Design evaluation is an important step during software development to ensure users' requirements are met. Crowd feedback represents an effective approach to tackling scalability issues of traditional design evaluation methods. Crowd-feedback systems are usually developed for a fixed use case and designers lack knowledge on how to build individual crowd-feedback systems by themselves. Consequently, they are rarely applied in practice. To address this challenge, we propose the design of a configuration system to support designers in creating individual crowd-feedback requests. By conducting expert interviews (N=14) and an exploratory literature review, we derive four design rationales for such configuration systems and propose a prototypical configuration system instantiation. We evaluate this instantiation in exploratory focus groups (N=10). The results show that feedback requesters appreciate guidance. However, there seems to be a trade-off between complexity and flexibility. With our research, we contribute with a generalizable concept to support feedback requesters to create individualized crowd-feedback requests to support scalable design evaluation for everyone.

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**.

KEYWORDS

configuration system, design feedback, crowdsourcing

ACM Reference Format:
Saskia Haug, Sophia Sommerrock, Ivo Benke, and Alexander Maedche. 2023. Scalable Design Evaluation for Everyone! Designing Configuration Systems for Crowd-Feedback Request Generation. In *Mensch und Computer 2023 (MuC '23), September 3–6, 2023, Rapperswil, Switzerland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3603555.3603566

MuC '23, September 3–6, 2023, Rapperswil, Switzerland

ACM ISBN 979-8-4007-0771-1/ 23/09…$15.00
https://doi.org/10.1145/3603555.3603566

1 INTRODUCTION

Continuous user involvement throughout the software development process is crucial for the success of software projects. One key activity in the user-centered design process is the continuous evaluation of software to ensure the fulfillment of functional and non-functional requirements. The drawbacks of the user-centered evaluation of software are high costs and scalability issues [24]. A solution for these challenges is crowd-feedback systems. These systems leverage crowdsourcing to collect large amounts of structured design feedback [27]. The focus of crowd-feedback systems is to collect explicit feedback on the perception of graphic designs (e.g., posters) [29], interactive prototypes [21], websites [22] or other types of software (e.g., chatbots) [4]. Research has shown that crowd-feedback systems can not only enable scalability and reduce costs as well as effort for feedback requesters [5, 21, 31] but can also produce reliable feedback that helps to improve the resulting designs [32]. However, crowd-feedback systems do not receive much adoption in practice. There are several potential causes for this. Crowd-feedback systems are usually fixed to specific use cases and provide limited flexibility for feedback requesters to adapt the system to their needs [11]. Especially designers with no development skills or limited methodological knowledge might have challenges applying crowd-feedback systems in practice. When creating crowd-feedback requests, many decisions need to be taken. Feedback requesters must not only decide which crowd to ask for feedback and how to incentivize it, but also what type of feedback is required (qualitative vs. quantitative), on which aspects the feedback is required (functional attributes vs. non-functional attributes vs. content), and how feedback providers shall be able to share their feedback (e.g., voice recording, collaboration, or markers) [11]. This determines what features of the crowd-feedback system need to be leveraged and in which way they should be combined. These challenges represent a large obstacle to the adoption of crowd-feedback systems and hinder the sufficient use of such systems. The research field of end-user development already extensively explored the need for tailoring software systems to individual requirements by domain experts without programming skills. The goal is there to make systems not only easy to use but also easy to develop [15]. Existing research also proposed design knowledge for configuration systems, however, mainly with a focus on the manufacturing or production domains [2]. Research on configuration systems to create individualized software is rather scarce. Thus, there is a need to investigate the specific context of designing configuration systems to enable feedback requesters to derive successful software evaluation strategies. Consequently, we articulate the following research question (RQ): *How to design a configuration system to support designers in creating effective customized crowd-feedback requests?*

To answer this RQ, we started by conducting 14 qualitative interviews with design experts to understand current issues regarding software evaluation. These interviews provided us with initial insights that we confirmed and extended with a literature review. Based on this, we developed four design rationales that our solution shall meet. We then instantiated them in a configuration system for crowd-feedback requests for customized software evaluation. Finally, we evaluated this artifact via an exploratory focus group workshop. With our work, we contribute to research by

- first, providing an understanding of designers' challenges of software evaluation
- second, proposing and evaluating four design rationales for the design of a configuration system for crowd-feedback requests, and
- third, developing a configuration system for an existing crowd-feedback system based on the proposed design rationales.

Individual design feedback requests will make crowd feedback applicable to a more diverse set of use cases and increase the integration of software evaluation in software development processes.

2 CONCEPTUAL FOUNDATIONS AND RELATED WORK

In this section, we provide conceptual foundations on design feedback in general and crowd-feedback systems in particular. Further, we provide an overview of related work on configuration systems.

2.1 Design Evaluation & Feedback

The evaluation of designs is the fourth step in the user-centered design (UCD) process [14]. The goal of the evaluation phase is to iteratively refine the design until the users' needs are met [3]. Traditional evaluation methods that build upon explicit user involvement include but are not limited to usability tests, interviews, and focus groups [3]. These methods are often fraught with scalability issues [24]. This general problem illustrates why continuous evaluation of designs with user involvement is often a major challenge in UCD [3]. In addition, designers often lack the required methodological knowledge to properly involve users in the evaluation process [18] and do not have access to a diverse set of (potential) users [17]. Running software is usually evaluated via feedback pop-ups or in dedicated feedback forums [1]. However, these approaches usually generate little feedback with low quality due to a lack of focus and structure [1]. Consequently, in practice, only a small percentage of development projects engage with users in every stage of the UCD approach [18]. To overcome some of these issues, crowd feedback has emerged as a new approach to collecting design feedback in a more scalable way. Crowd feedback originates in the graphic design domain as an alternative to peer feedback [32]. Dedicated crowd-feedback systems offer feedback providers structure that increases the feedback quality [16]. Another benefit of crowd feedback is that designers do not necessarily need access to real users anymore but can also use an anonymous crowd as it can be found on platforms like Amazon Mechanical Turk (MTurk). Still, crowd feedback can also be collected from stakeholders or potential users [6, 22]. In the next section, we present what dedicated crowd-feedback systems look like and what new challenges these systems bear.

2.2 Crowd-Feedback Systems

Crowd-feedback systems present an approach to solving the scalability issues related to user involvement in software development. These systems collect large amounts of structured feedback by engaging a group of humans, which can be but not must be real or potential users [11]. In research, various studies on crowd-feedback systems exist. A big advantage compared to simple surveys as they can be created with tools like LimeSurvey or Google Forms is that crowd-feedback systems usually enable the feedback collection while the feedback provider is actually using the system (e.g., [9, 30]. Crowd-feedback systems can also apply dedicated features that are not available in online survey tools like markers to pin comments to a specific element [10]. According to Haug and Maedche [11], the core element of crowd-feedback systems are their design characteristics and the selected crowd configuration. The crowd configuration describes what type of crowd is asked for feedback (anonymous, users, students, and convenience) and how this crowd is incentivized (money, involvement and improvement, credits, social compensation, and gamification). The design characteristics are split into feedback collection mechanisms and interactivity cues. While feedback collection mechanisms (questionnaire, free text field, categories, selection, and direct manipulation) conceptualize all features to collect feedback, the interactivity cues (collaboration, markers, context, and recording) describe additional features that help feedback providers to improve their feedback quality or enrich their feedback [11]. While researchers have demonstrated that crowdsourcing high-quality feedback is feasible with dedicated crowd-feedback systems, there are still discussions about definitions and measures for design feedback quality [9]. Also, feedback might be less honest when people are paid for providing feedback [9].

Most of the existing crowd-feedback systems have focused on evaluating static designs like posters. For example, Voyant is a popular example of a crowd-feedback system with the goal of collecting feedback on graphic designs [29, 30]. It captures the crowds' first impressions and how well specific goals and design guidelines are met. There also exist a few systems that evaluate interactive designs like chatbots, mock-ups, or running websites. Many of the studies focused on achieving a high feedback quality or optimizing the resulting designs. Thereby, the applicability of these systems in practice was often neglected [10]. Consequently, all existing crowd-feedback systems in research are fixed to a specific use case or to the evaluation of a specific software system [11]. One system that allows the evaluation of interactive designs and combines multiple design characteristics of crowd-feedback systems to understand how they are perceived by users is Feeasy [10]. Feeasy collects feedback via a free text field, categories, and star ratings that are attached to the categories. Additionally, it contains markers, context, and recording, in the form of a speech-to-text feature as interactivity cues. Haug et al. [10] used Feeasy to understand the effect of the design characteristics of crowd-feedback systems on the resulting feedback quality and quantity. Their results can serve as a basis to better support designers in creating individual crowd-feedback requests and tailoring the design to the respective requirements and context. For an extensive overview of existing crowd-feedback systems, we recommend the literature review of Haug and Maedche [11].

2.3 End-User Development and Configuration Systems

End-user development (EUD) shall empower software users without a background in programming to develop or modify their own software systems [15]. This allows for more flexible and tailored use of these applications. Research in the field of EUD proposes methods, techniques, and tools for creating, modifying, or extending software artifacts [15]. As in our case, the designers who shall configure the crowd-feedback systems are not the actual end-users of the crowd-feedback system, we will consider EUD only as a side topic in the design of our configuration system.

Regarding configuration systems, there mainly exist two types: needs-based and parameter-based [23]. Needs-based configuration systems ask users to specify the relative importance of their needs regarding the resulting product. An algorithm then combines the design parameters so that the user's needs are matched as closely as possible. Parameter-based configuration systems on the other hand allow users to directly specify the design parameters for the resulting product. Therefore, these systems are usually more flexible but also require more expertise from users [23]. Research on configuration systems for adapting software to users' needs is rather scarce. Feine, Morana, and Maedche [8] designed a chatbot social cue configuration system. This system supports chatbot engineers in accessing descriptive knowledge to make more justified social cue design decisions by transforming the descriptive knowledge into prescriptive knowledge. While this configuration system is applying a needs-based approach by providing recommendations based on the target user, task, and context, we will aim for a parameter-based approach. This is because the descriptive knowledge of design features of crowd-feedback systems and their effects is rather limited and therefore no justified design decision can be automatically derived purely based on users' needs. Parameter-based configuration systems are characterized by many decisions that users must make. So, on a more abstract level, the design of crowd-feedback requests can be seen as a series of consecutive decision tasks in which the feedback requester is the advice taker, and the configuration system serves as the advice giver. In this analogy, selecting a feature or a type of crowd can be seen as a single decision task.

3 DESIGNING A CONFIGURATION SYSTEM FOR FEEDBACK REQUEST CREATION

To design a configuration system that can effectively support designers in creating individual crowd-feedback requests, we combined insights from expert interviews with an exploratory literature review. In the first step, we focused on collecting the fundamental requirements of designers and product owners and understanding how current solutions need to be improved to meet these requirements to develop design rationales for our system design. In the next step, we developed a software prototype based on these design rationales.

3.1 Interview Study and Literature Review

We interviewed fourteen design experts to understand what issues they experience when evaluating prototypes or software and what the related processes look like. The design experts (64.29% female) included UX designers, UX managers, UX researchers, and product owners. The interviewees were on average 36.29 years old (SD = 9.96 years), mainly worked in large companies, and had on average 10.71 years (SD = 8.03 years) of work experience. The interviews took on average 24.51 min (SD = 4.92 min). The interviews were conducted in German, then transcribed, translated, and coded following an empirical-to-conceptual approach, mainly focusing on designers' issues related to design evaluation. To verify our results and to get a broader picture of designers' challenges when evaluating designs and potential solutions, we conducted an exploratory literature review on existing crowd-feedback systems and the related feedback processes. We used publications from a literature review on the state-of-the-art of crowd-feedback systems by Haug and Maedche [11] as a starting point and extended our search based on their papers. To identify and structure the issues, we coded all 21 papers that we found relevant following an iterative empirical-to-conceptual approach [20]. This also helped us to assess the importance of identified issues based on their number of occurrences in different papers. In the following, we bring together the insights from the interviews and learnings from the literature.

Regarding the implementation of software evaluation practices, we identified four core problem areas based on our interviews: budget and time, process and methodology, internal collaboration, and diversity of participants. Exemplary quotes for these problems are summarized in Table 1.

When looking at the literature, we learned that many of these challenges could be solved by applying crowd-feedback systems. Crowd feedback offers a scalable approach to collecting feedback from a diverse group of people. Accordingly, we wondered, why crowd feedback is not applied in practice and took a closer look at the challenges that potential feedback requesters might face. From the analysis of existing literature, we learned that the preparation of crowd-feedback requests is experienced as time-consuming and effort-intensive [13, 25]. Especially for requesters with little technical experience and skills, the creation of a crowd-feedback request is a complex challenge [21]. Requesters with little methodological knowledge also find it difficult to learn the necessary techniques and skills to create a feedback request [13, 18].

Decisions on crowd-feedback system features and settings are complex and hard to make without specific knowledge. For example, research shows that the choice of the crowdsourcing platform that is used for the crowd-feedback request affects the received feedback [31]. Paid task markets are found to provide feedback with more design suggestions while responses from web forums lead to more process-oriented feedback [31]. Also, most incentives are linked to the type of crowd used. For example, the anonymous crowd is found to be mostly financially incentivized [11]. Also, the features

Table 1: Results of expert interviews

Problems with design evaluation in practice	Quotes
Budget and time	*"Very often, strict evaluation is simply dispensed with because of the costs involved, and people say they'll go live and try to fix things with the live feedback." (E.1)* *"That is also the biggest pain point of my colleague or of the UX people [...], that in many projects this phase comes too short. Be it budget or be it time." (E.3)* *"Of course, that would be very nice if you still had the time and budget to really involve the user in the entire process. Well, that's often simply not possible, also from the client's point of view." (E.1)*
Process and methodology	*"That's actually always the challenge in the enterprise setting, or in the business UX field, that you have the right people in place at the right time." (E.6)* *"This [the evaluation process] is relatively unstructured, it is not like we have a questionnaire or somehow collect structural user feedback." (E.9)*
Internal collaboration	*"Because you are somewhere dependent on the product owner, you can't go out on the street yourself, you have to coordinate with the product owner somehow and often I have the feeling that this is not so important to them because they always say that they know what the end user needs." (E.3)*
Diversity of participants	*"If I do then I might ask 3-4 people and the feedback on the prototypes is then based on the feedback from those three to four people." (E.11)* *"And that's where I definitely still see a big gap [...] that we still have a lot of only internal feedback and, above all, only feedback from people who are involved there anyway." (E.9)*

of crowd-feedback systems need to be adapted to the goal of the feedback study and to the perceptions of feedback providers [10]. Supplementary, one must consider that different end-users have different design needs [16]. For instance, the support for structured feedback can be reduced as soon as the crowd users gain some experience [22]. Existing crowd-feedback systems are found to have a fixed set of design characteristics [11]. Consequently, they are fixed to one specific evaluation and are not adaptable to other use cases [16]. The inflexibility of crowd-feedback systems is further illustrated by the findings that they are mostly fixed to the evaluation either during the development or operation phases [11]. Only a few systems focus on the evaluation of interactive systems [11]. This means that existing crowd-feedback systems have usually a fixed set of design characteristics and can hardly be reused for other use cases.

Based on these insights, our goal was to develop a configuration system that shall help design experts without knowledge of crowd feedback to create individual crowd-feedback requests for their design projects.

3.2 Design Rationales

With respect to the goal of our study, we synthesized the findings of our initial interviews and literature review to develop four design rationales (DRs) for our system design:

(1) *User Guidance.* Users of our system might have varying methodological knowledge and technical skills. Therefore, our system needs appropriate functions to guide and support feedback requesters during the feedback request creation process so that they can quickly and easily create a crowd-feedback request. This includes the selection of crowd-feedback system features, but also the distribution of feedback requests.

(2) *Effect of Design Characteristics.* For designers without any experience in crowd feedback, it is not possible to comprehend the effects of individual design decisions regarding the feedback request. Our system shall display the potential effects of feature selections on the resulting feedback so that feedback requesters can select and combine appropriate features according to their current requirements.

(3) *System Customization.* Every design project and therefore every feedback request is individual. Our system needs functions to customize feedback requests so that feedback requesters can collect feedback for different tasks and use cases.

(4) *Crowdworker Perspective.* We want crowdworkers to be able to submit high-quality feedback when they are using the configured crowd-feedback system. Therefore, our configuration system needs to have functionalities that allow the feedback requester to consider the crowd's needs and requirements during the configuration process.

3.3 System Design

We developed a prototype based on our four design rationales. For this, we combined a real software artifact with a Figma design mock-up. This allows us to reduce the development effort before verifying the design rationales are valid and to quickly iterate the design. The selection process of design features of the configuration is thereby implemented in the software prototype, while the platform which feedback requesters can use to manage their requests is implemented as a design mock-up in Figma. We aimed to keep our design of the configuration system very flexible, so that it can be used for a diverse set of designs including interactive prototypes or mock-ups, but also static designs like posters. Figure 1 shows our software artifact and a final feedback request. The colored boxes indicate how our four design rationales were transferred to design features. Feedback requesters start in the *Projects* tab in the configuration system. There, they can manage their feedback projects and related feedback requests. Additionally, the platform contains an *Apps* tab that presents different crowdsourcing platforms and their characteristics, including the advantages and disadvantages of each platform, and supports designers in deciding how to distribute their feedback requests. The third tab in the menu is the *Help Center*. There, users can get support for interacting with the configuration system, as well as learn about descriptive knowledge for feedback requests in the form of guidelines. For each design project, users must upload their design in the form of one or multiple files. These files can be images for static designs or multiple HTML files for interactive designs. Then, they can create a new feedback request. The step-by-step configuration of the feedback request is implemented as a software artifact.

Based on the core features of Feeasy [12] and additional feedback features according to Haug et al. [11], users are guided through five configuration steps as displayed in Figure 2. First, feedback requesters can choose if they want to offer feedback providers a context in the form of a scenario and add text to each step in the scenario. This can help feedback providers especially, when feedback is collected on an interactive prototype to navigate through it. Then, they can choose if they want to include a questionnaire either with answers in the form of scales or with text entries. The third step allows them to add a free text field and related features, such as categories with a star rating, markers, and a recording feature. In the fourth step, users can decide if they want to include a collaboration feature that enables feedback providers to see the comments of others and react to them. Finally, feedback requesters can decide if they want to give users a predefined time after which they are allowed to submit their feedback. During the whole configuration process, a panel on the right side of the screen shows users how implementing the respective feature would look. For each design option, a tooltip with a definition of the feature is next to the question and a box with descriptive knowledge of the design option is below the headline. For some features, the configuration system offers recommendations, e.g., categories that are frequently used in research. Feedback requesters are also able to customize the feedback features, for example, add individual categories. The user interface for the configuration process also contains a feature count that shows users how many features they have already combined. This shall help them to keep the perspective of the feedback providers in mind and not overload the feedback request with unnecessary features. When feedback requesters are done with their configuration process, they get redirected to the projects tab on the configuration platform. There, a link is created that leads feedback requesters to the configured feedback request. The idea is that feedback requesters can then share this link with their desired crowd or integrate it into a survey.

4 EVALUATION STUDY

To confirm our design rationales and understand how potential feedback requesters perceive our configuration system, we conducted an evaluation in the form of two exploratory focus group workshops. We decided on an in-person workshop instead of following a crowdsourcing approach as our goal was to directly interact with potential users and be able to react to their feedback, questions, and ideas immediately. In the following, we present the evaluation procedure and describe our participants.

4.1 Procedure

Our configuration system shall work for all types of graphic and interactive designs. Therefore, we conducted two workshops. While the first workshop was focusing more on evaluating interactive designs, such as website prototypes, the second one addressed the evaluation of static designs like posters or simple mock-ups. After a short introduction to crowd-feedback systems and crowd-feedback requests in general, we provided the participants with an overview of the steps in our configuration system. We then showed them an exemplary configuration for creating a feedback request for the evaluation of a simple website prototype (workshop 1) or a poster design (workshop 2). The demonstrating researcher presented the use case and showed which decisions a feedback request creator must make. After the demonstration, the open questions of the participants were clarified. Then, each participant had ten minutes to try out the configuration system him/herself. In the next step, we explained the Strength-Weakness-Opportunity-Threat (SWOT) analysis method to the participants that we used to structure the exploratory focus group. The participants then had time to write down their perceived strengths, weaknesses, opportunities, and threats of the configuration system. Finally, the results were read out loud and explained by the participants. Other participants and the researchers could discuss ideas or ask follow-up questions to clarify each point. In the second workshop, the moderating researcher presented the results of the first workshop, initiating a discussion about the similarities and differences between the results. With this input, participants had the opportunity to expand their SWOT analysis.

Both workshops were recorded and the final results in the SWOT matrix were captured. As the workshops were conducted in German, the recordings were first transcribed and then translated into English. Then, one of the authors analyzed and coded the SWOT results and transcriptions regarding the four design rationales.

4.2 Participants

Exploratory focus groups propose improvements to refine the design [26]. As our configuration system shall be designed for different user types and the evaluation of different types of designs, we aimed for a diverse set of participants. This is also consistent with the recommendation to mix different skill sets for the evaluation of decision-aid tools [26]. Focus groups should involve between four and twelve participants [19]. Therefore, we invited ten participants with various backgrounds and levels of experience to participate in our evaluation (nine male, MAge = 31.1 years, MDesignExp = 8.4 years, MDesignEvalExp = 4.8 years) and split them according to

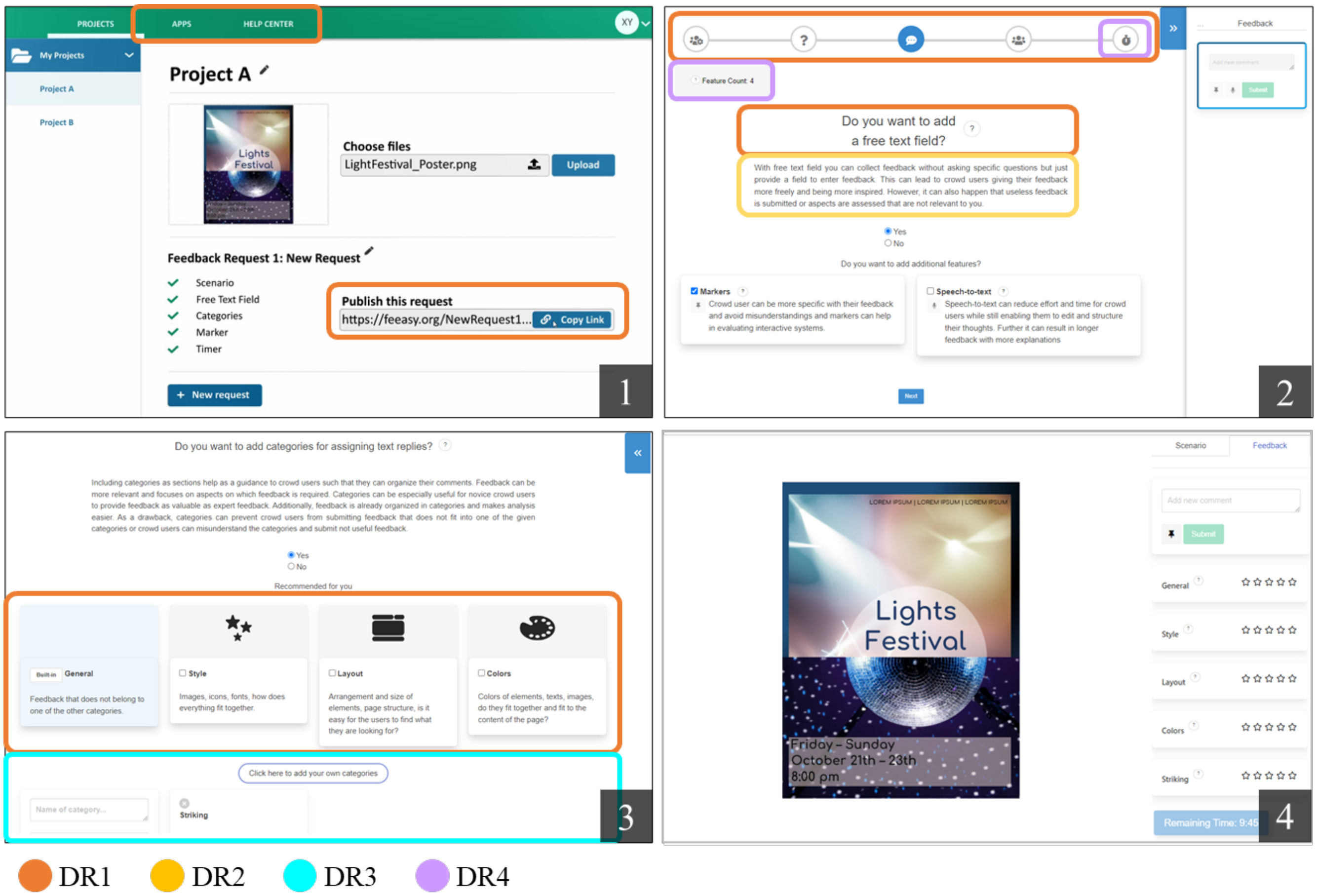

Figure 1: Configuration system artifact as an instantiation of our four design rationales. (1) For each project, users must add their design via a file upload. (2) When creating a new feedback request users are guided through each step. On top they have a progress bar with five steps, on the right side, they see how each feature would be implemented in their feedback request. (3) After deciding on a feature, users can configure the feature. (4) The final feedback request shows the uploaded design and the feedback panel.

their design background in the two workshops: P.1 to P.6 participated in Workshop 1 and P.7 to P.10 in Workshop 2. Details on our participants are shown in Table 1.

5 RESULTS

We analyzed the results of the SWOT analysis according to our four design rationales. The most important results for each design rationale are summarized in Table 2 and explained in more detail below.

In general, our participants appreciated the user guidance provided by the configuration system. This was explained by P.6 as follows: *"I think the user guidance is very well done. I had the feeling that even though I'm seeing it for the first time, I always know where I am, what I'm doing, how to continue, how to get back"*. In addition, participants expressed that the system is *"fast and simple"* (P.8) and allows them to quickly create a feedback request (*"I could quickly upload every little piece and get rapid feedback to evaluate it the next day"*, P.8). However, at the same time, the configuration system was still perceived as very complex and *"... the look and clarity of what exactly is being applied could be slightly [...] enhanced from the user guidance perspective"* (P.5). One reason for the high perceived complexity is that *"... you are flooded with a lot of text"* (P.10). P.9 explicates: *"Quite a lot of text at once and the text itself not that appealing to read"*. Therefore, participants recommended *"... hide out the points that you might understand, so you can sort of pop in what you are interested in to get more information, in a popup window for example"* (P.10). Further, participants would have liked to receive more guidance in selecting the right crowd and appropriate incentive as they perceive this as a major problem in feedback requests. The integration of research insights into the configurator was perceived as a major strength of our configuration system: *"Particularly for inexperienced users, I found it good that the positive and negative effects of features were described in the description"* (P.4). Concerning the explanations in the information text on the design features, it was noted that a learning effect of the

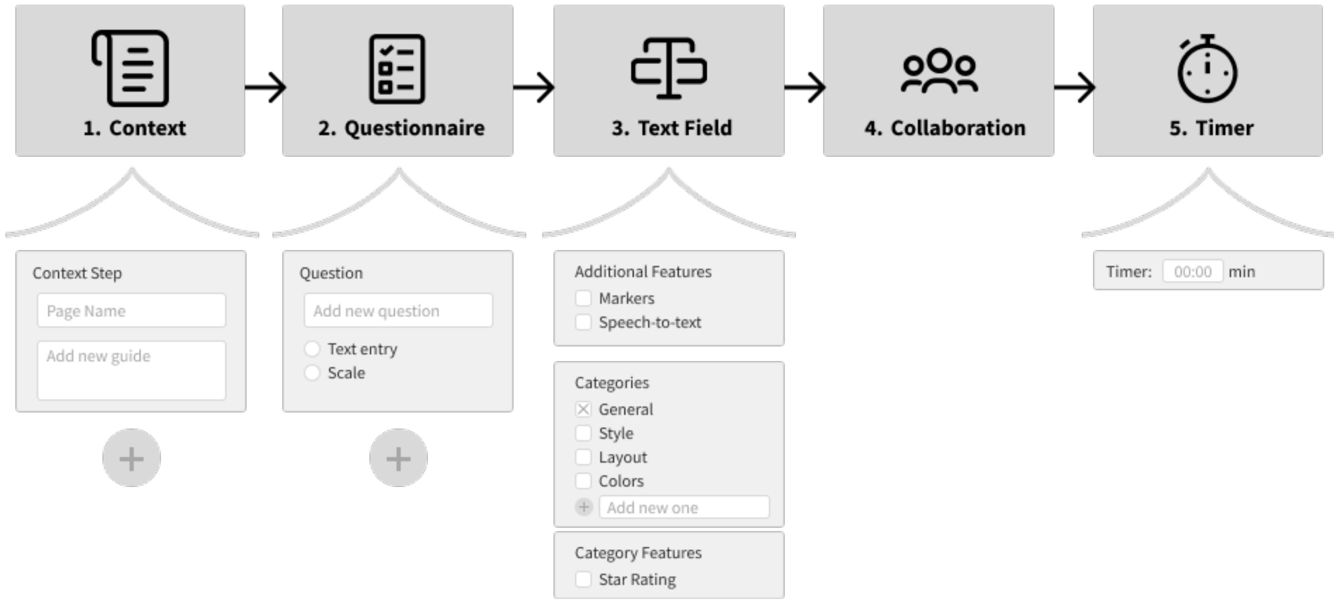

Figure 2: Configuration process as it is implemented in our instantiation of the configuration system. First, designers can add context to their design. In multiple steps, they can explain the design and guide feedback providers through the interaction with it. In the second step, designers can add a questionnaire and enter questions that can be answered via text entry or a 5-point Likert scale. If designers decide to add a free text field in the third step, they can also choose additional features, such as markers, speech-to-text functionality, categories, and star ratings. The fourth step allows designers to add a collaboration feature. And finally, in the fifth step, designers can add a timer to limit the time users have to provide feedback.

Table 2: Demographic information on participants of the focus group workshops

ID	Profession	Gender	Age	Experience in years		Level of Design Knowledge
				Design	Design Evaluation	
P.1	Experience Manager	M	39	20	20	Very high
P.2	UX Designer	F	24	2	3	Medium
P.3	IT Specialist	M	44	12	8	Medium
P.4	IS Student	M	26	0	0	Very low
P.5	Web Designer	M	28	8	7	High
P.6	UX Developer	M	29	4	1	Medium
P.7	Graphic Designer	M	42	23	0	Very high
P.8	Graphic Designer	M	27	7	3	High
P.9	PhD Student in IS/HCI	M	28	4	2	High
P.10	Media Communications Designer	M	24	4	4	High

users is not taken into account. Some information is only relevant for the first usage, *"... which you would then probably not require the second time anyway"* (P.8). Participants proposed to include even more recommendations, also based on design experts' experiences. Regarding our third design rationale, the configurability of the feedback requests, participants were indecisive if they liked the flexibility and the openness or if it was too much for them: *"It [the configuration system] seems to be flexible. I had the feeling that I could go into a lot of detail and do a lot of things. On the one hand, I think that's good, but on the other hand, it's not good"* (P.1). Some participants requested having even more design features included that they could configure in their feedback requests. Lastly, we integrated the perceptions of the crowd in the form of the feature count, this led to participants always having the perception of the crowd in their view. One participant reflected on this as follows: *"I think the count of features at the end is pretty cool. Because you can see if it becomes too crowded when you give feedback. I think that makes sense"* (P.9). However, it was not clear to all participants how to consider the crowd's perception in the form of the feature count in their feedback request. One participant stated that s/he wasn't sure *"...what that meant for my specific request. So, I wasn't clear on whether I did it well or not"* (P.2).

Table 3: Summary of the results of the SWOT analysis according to design rationales (DRs)

DR	Strengths	DR	Weaknesses
1	S1. Feedback requesters are guided through the configuration process simply and efficiently.	1	W1. Some parts of the configuration process lack clarity and are very complex.
2	S2. The effects of different design options are sufficiently explained, especially when users are inexperienced or do not know their goal.	2	W2. The learning effects of users are not considered in the design.
3	S3. The feedback request is very flexible and can be adapted to different use cases.	3	W3. The configuration system and the options are at some points too open.
4	S4. The perspective of the crowd is always in view via the feature count.	4	W4. The connection between decisions and the meaning regarding the crowds' perception was sometimes not obvious.
DR	**Opportunities**	**DR**	**Threats**
1	O1. Offering templates could simplify and speed up the configuration process.	1	T1. The incentivization for feedback providers should not be neglected.
2	O2. Including experts' recommendations would help in making the right configuration decisions.	3	T2. Users could get lost in the flexibility of the system.
3	O3. The configuration system could offer even more design features.		

6 DISCUSSION

While crowd-feedback systems offer feedback requesters a scalable and effective way to collect feedback on graphic designs, prototypes, and software designs, existing crowd-feedback systems do not support feedback requesters in creating and configuring individual feedback requests. To address this problem, we designed a configuration system that supports feedback requesters in individual crowd-feedback request creation. Drawing on existing research on challenges in design evaluation, the effects of feedback features, and decisional guidance, we examined how the combination of descriptive knowledge and a step-by-step configuration process can support feedback requesters in creating individual feedback requests. Subsequently, we conducted an exploratory focus group evaluation with a diverse group of designers. The results of our evaluation show that our approach has the potential to solve existing problems of designers, while there are still more ideas that could be explored in future research. We learned that offering too many design options makes designers feel lost in the process. Also, they emphasized the idea of offering templates based on experts' recommendations. Therefore, our study provides valuable theoretical contributions and practical implications that we discuss in the following.

6.1 The crowds' perspective still needs more attention

We did not receive as much feedback on DR4 as on the other three DRs. This might be caused by feedback requesters still not having the crowds' perceptions in mind when creating feedback requests. We instantiated this rationale mainly via the feature count. One main issue with the feature count was that participants felt that it was not integrated sufficiently into the prior decisions. It was not clear enough how users shall consider the feature count when making their design decisions. Further, we interpret the lack of feedback on DR4 so that feedback requesters do not think about the crowds' perspective much as they assume that the configuration system automatically respects their needs. Consequently, in further iterations, we need to come up with alternative instantiations of DR4 to ensure the crowd's perception is always considered during the configuration process.

6.2 There is a trade-off between flexibility and complexity of the configuration system

Two core topics during the focus group evaluation were the flexibility of the configuration system due to the combination of many feedback features and the high complexity of the configuration process due to much information and many interdependencies that must be considered. This issue demonstrates tensions between our design rationales, especially DR1 and DR3. On the one hand, feedback requesters need guidance, on the other hand, they want a flexible configuration system so that they can completely customize each feedback request. We believe that every configuration system needs to find a balance between flexibility and complexity. Although both aspects are not two extremes on a shared scale, they need dedicated features to be compatible as often one comes at the price of the other. While needs-based configuration systems are less complex for users, they provide less flexibility and transparency [23]. The lack of transparency was also a major concern in the evaluation of the configuration system of Feine et al. [8]. We wanted to counteract this problem by offering a parameter-based configuration system. This was also necessary because there is still a lack of knowledge of the effects of feedback features. Crowd-feedback systems are complex systems with multiple features to combine. To reduce the complexity of our configuration system, many participants suggested offering templates for specific use cases that can then be adapted by feedback requesters. This would be a hybrid approach of the needs-based and parameter-based systems. These templates could be based on the three research streams identified by Haug and Maedche [11]: anonymous crowd-feedback, real user crowd-feedback, and hybrid crowd-feedback. By offering a template, users would need to make one decision when choosing a template and can then make additional fine-tuning decisions but they are not forced anymore to decide for or against every single feedback feature. The templates could be developed by experts considering theoretical knowledge and practical implications of feedback features. The templating approach is consistent with the recommendation of Weinmann et al. [28] who suggested offering a hybrid approach for configuration systems, also for users with different levels of expertise. An alternative to offering templates by the system could be allowing users to share their requests and reuse feedback requests of other users. By rating or commenting on feedback requests of others, templates could be created organically by the users of the configuration system. This would also bring the system closer to the idea of end-user development that we mentioned earlier.

When offering templates, another approach could be to implement them in established survey tools (e.g., LimeSurvey) similarily to QButterfly [7], a toolkit for conducting usability studies in LimeSurvey or Qualtrics. Thereby, we can achieve the same benefits: reduce authoring time and complexity, empower users without programming skills to conduct design studies facilitate the re-use of the existing functionality of these tools, and facilitate the replication of ideas.

6.3 Experienced users of the configuration system might need advanced functionalities

We explicitly decided to design our configuration system for both, novice and expert feedback requesters. Therefore, some features of the configuration systems might be more useful for novices, while others are specifically designed for experts. Parameter-based configuration systems need users usually to be experienced in the specific domain to make the right decisions [23]. As we did not assume that all users already know which are the right decisions for their use case, we focused much on user guidance by explaining the advantages and disadvantages of feedback features. However, one thing we did not consider sufficiently here was the learning effects of users. This point was raised during the evaluation of our configuration system. When users have understood the effects of the design features, they do not need to read the information on advantages and disadvantages again every time they want to create a new feedback request. Therefore, participants requested to adapt the UI more to experienced users of the configuration system by allowing them to hide information texts or store user inputs for the following configuration processes.

6.4 Limitations and Future Research

Of course, there are also limitations in our work that need to be considered. First, participants in our focus group followed an artificial use case when they tested the prototype. These limitations might have biased participants' perceptions of our system. In future work, we want to develop a completely functional artifact. This will make the evaluation results even more insightful and reliable. Second, we used an exploratory focus group to perform a qualitative evaluation of the configuration system artifact. The goal of the evaluation was to understand if the configuration system actually supports feedback requesters in creating design feedback requests and how we can further improve the system. While this approach enabled us to collect valuable insights into users' interaction with the configuration system and innovative ideas, future research should conduct a quantitative evaluation to understand to what extent the configuration system is usable and helpful for feedback requesters.

Overall, our design rationales showed to be key for providing feedback requesters the possibility to create individual crowd-feedback requests. They are partly contradictory, which means that a good balance between them, especially the flexibility and complexity of the configuration system, needs to be found. Our evaluation also sheds light on additional design issues, which offer valuable starting points for further improvements of the configuration system for crowd-feedback requests in upcoming design cycles.

7 CONCLUSION

While crowd-feedback systems offer a scalable way to collect design feedback, they, however, do not support feedback requesters (e.g., designers) in creating and configuring individual feedback requests. Therefore, in this paper, we present a study on the design of configuration systems that support feedback requesters in individual crowd-feedback request creation. Drawing on existing research on challenges in design evaluation and an interview study with experts, we contribute with four design rationales to support feedback requesters in selecting and configuring feedback features while considering the effects of feedback features on the crowds' perceptions as well as on the feedback quality and quantity. Our results show that feedback requesters appreciated the guidance but leaving too many decisions open made them feel lost in the process. Also, they emphasized the idea of offering templates based on experts' recommendations. Overall, our study contributes design knowledge that can be applied to guide feedback requesters through the decision-making process of creating crowd-feedback requests for the evaluation of software designs. With this, we contribute to making software evaluation more simple, scalable, and efficient and support the development of more human-centered software.

REFERENCES

[1] M Almaliki, C Ncube, and R Ali. 2014. The Design of Adaptive Acquisition of Users Feedback: An Empirical Study. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*. IEEE, New York, NY, USA, 1–12. https://doi.org/10.1109/RCIS.2014.6861076

[2] Preethu Rose Anish and Smita Ghaisas. 2014. Product Knowledge Configurator for Requirements Gap Analysis and Customizations. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. Institute of Electrical and Electronics Engineers Inc., 437–443. https://doi.org/10.1109/RE.2014.6912295

[3] Manuel Brhel, Hendrik Meth, Alexander Maedche, and Karl Werder. 2015. Exploring Principles of User-Centered Agile Software Development: A Literature Review. *Information and Software Technology* 61 (2015), 163–181. https://www.sciencedirect.com/science/article/pii/S0950584915000129

[4] Yoonseo Choi, Toni-Jan Jan Keith Palma Monserrat, Jeongeon Park, Hyungyu Shin, Nyoungwoo Lee, and Juho Kim. 2021. ProtoChat: Supporting the Conversation Design Process with Crowd Feedback. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW* 4, CSCW3 (2021), 19–23. https://doi.org/10.1145/3432924

[5] Steven Dow, Elizabeth Gerber, and Audris Wong. 2013. A Pilot Study of Using Crowds in the Classroom. In *Proceedings of the 2013 Conference on Human Factors in Computing Systems*. Association for Computing Machinery (ACM), New York, NY, USA, 227–236. https://doi.org/10.1145/2470654.2470686

[6] Matthew W Easterday, Daniel Rees Lewis, and Elizabeth M Gerber. 2017. Designing Crowdcritique Systems for Formative Feedback. *International Journal of Artificial Intelligence in Education* 27, 3 (2017), 623–663. https://doi.org/10.1007/s40593-016-0125-9

[7] Nico Ebert, Björn Scheppler, Kurt Alexander Ackermann, and Tim Geppert. 2023. QButterfly: Lightweight Survey Extension for Online User Interaction Studies for Non-Tech-Savvy Researchers. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3580780

[8] Jasper Feine, Stefan Morana, and Alexander Maedche. 2019. Designing a Chatbot Social Cue Configuration System Digital nudging View project Designing a Process Mining-Enabled Decision Support System for Business Process Transformation View project. In *Proceedings of the 40th International Conference on Information Systems (ICIS)*. Association for Information Systems (AIS), New York, NY, USA, 17. http://ksri.kit.edu

[9] Saskia Haug, Ivo Benke, Daniel Fischer, and Alexander Maedche. 2023. CrowdSurfer: Seamlessly Integrating Crowd-Feedback Tasks into Everyday Internet Surfing. In *CHI '23: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3544548.3580994

[10] Saskia Haug, Ivo Benke, and Alexander Maedche. 2023. Aligning Crowdworker Perspectives and Feedback Outcomes in Crowd-Feedback System Design. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW (2023), 1–28. https://doi.org/10.1145/3579456

[11] Saskia Haug and Alexander Maedche. 2021. Crowd-Feedback in Information Systems Development: A State-of-the-Art Review. In *Proceedings of the 42nd International Conference on Information Systems (ICIS) 2021*. Association for Information Systems (AIS), New York, NY, USA, 17. https://aisel.aisnet.org/icis2021/is_design/is_design/4

[12] Saskia Haug and Alexander Maedche. 2021. Feeasy: An Interactive Crowd-Feedback System. In *Adjunct Publication of the 34th Annual ACM Symposium on User Interface Software and Technology, UIST 2021*. Association for Computing Machinery (ACM), New York, NY, USA, 41–43. https://doi.org/10.1145/3474349.3480224

[13] Julie S. Hui, Michael D. Greenberg, and Elizabeth M. Gerber. 2014. Understanding the Role of Community in Crowdfunding Work. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 62–74. https://doi.org/10.1145/2531602.2531715

[14] ISO 9241-210. 2019. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. , 21 pages. https://www.iso.org/standard/77520.html

[15] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. 2006. End-User Development: An Emerging Paradigm. In *End User Development*. Springer Netherlands, 1–8. https://doi.org/10.1007/1-4020-5386-x{_}1

[16] Kurt Luther, Jari Lee Tolentino, Wei Wu, Amy Pavel, Brian P Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P Dow. 2015. Structuring, Aggregating, and Evaluating Crowdsourced Design Critique. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. Association for Computing Machinery (ACM), New York, NY, USA, 473–485. https://doi.org/10.1145/2675133.2675283

[17] Xiaojuan Ma, Yu Li, Jodi Forlizzi, and Steven Dow. 2015. Exiting the Design Studio: Leveraging Online Participants for Early-Stage Design Feedback. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. Association for Computing Machinery (ACM), New York, NY, USA, 676–685. https://doi.org/10.1145/2675133.2675174

[18] Ji Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. 2005. The state of user-centered design practice. *Commun. ACM* 48, 3 (3 2005), 105–109. https://doi.org/10.1145/1047671.1047677

[19] David L. Morgan. 1997. *Focus Groups as Qualitative Research*. SAGE Publications, Inc. https://doi.org/10.4135/9781412984287

[20] Robert C Nickerson, Upkar Varshney, and Jan Muntermann. 2013. A method for taxonomy development and its application in information systems. *14769344* 22, 3 (2013), 336–359. https://doi.org/10.1057/ejis.2012.26

[21] Jonas Oppenlaender, Elina Kuosmanen, Andrés Lucero, and Simo Hosio. 2021. Hardhats and Bungaloos: Comparing Crowdsourced Design Feedback with Peer Design Feedback in the Classroom. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery (ACM), New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445380

[22] Jonas Oppenlaender, Thanassis Tiropanis, and Simo Hosio. 2020. CrowdUI: Supporting Web Design with the Crowd. *Proceedings of the ACM on Human-Computer Interaction* 4, EICS (2020), 1–28. https://doi.org/10.1145/3394978

[23] Taylor Randall, Christian Terwiesch, and Karl T. Ulrich. 2007. Research Note—User Design of Customized Products. *Marketing Science* 26, 2 (3 2007), 268–280. https://doi.org/10.1287/mksc.1050.0116

[24] Jean Scholtz. 2001. Adaptation of Traditional Usability Testing Methods for Remote Testing. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*. IEEE Computer Society, Los Alamitos, CA, USA, 5030. https://doi.org/10.1109/HICSS.2001.926546

[25] R Snijders, F Dalpiaz, S Brinkkemper, M Hosseini, R Ali, and A Ozum. 2015. REfine: A Gamified Platform for Participatory Requirements Engineering. In *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*. IEEE, 1–6. https://doi.org/10.1109/CrowdRE.2015.7367581

[26] Monica Chiarini Tremblay, Alan R ; Hevner, Donald J Berndt, Monica Chiarini Tremblay, ; Chiarini, and Alan R ; Hevner. 2010. Focus Groups for Artifact Refinement and Evaluation in Design Research. *Communications of the Association for Information Systems* 26, 1 (6 2010), 27. https://doi.org/10.17705/1CAIS.02627

[27] Helen Wauck, Yu Chun Yen, Wai Tat Fu, Elizabeth Gerber, Steven P Dow, and Brian P Bailey. 2017. From in the class or in the wild? Peers provide better design feedback than external crowds. *Conference on Human Factors in Computing Systems - Proceedings* 2017-May (2017), 5580–5591. https://doi.org/10.1145/3025453.3025477

[28] Markus Weinmann, Martin Hibbeln, and Susanne Robra-Bissantz. 2011. Customer-Oriented Configuration Systems: One Type Fits All?. In *19th European Conference on Information Systems, ECIS 2011*. Association for Information Systems (AIS), New York, NY, USA. https://aisel.aisnet.org/ecis2011/132

[29] Anbang Xu and Brian P Bailey. 2014. A System for Receiving Crowd Feedback on Visual Designs Abstract. In *CSCW Companion '14: Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. Association for Computing Machinery (ACM), New York, NY, USA, 37–40.

[30] Anbang Xu, Shih-Wen Wen Huang, Brian P. Bailey, and Alejandra Duque-Estrada. 2014. Voyant: Generating Structured Feedback on Visual Designs Using a Crowd of Non-Experts. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing* 24, 74 (2014), 1433–1444. https://doi.org/10.1145/2531602.2531604

[31] Yu Chun Grace Yen, Steven P. Dow, Elizabeth Gerber, and Brian P. Bailey. 2016. Social network, Web Forum, or Task Market? Comparing Different Crowd Genres for Design Feedback Exchange. In *DIS 2016 - Proceedings of the 2016 ACM Conference on Designing Interactive Systems: Fuse*. Association for Computing Machinery (ACM), New York, NY, USA, 773–784. https://doi.org/10.1145/2901790.2901820

[32] Alvin Yuan, Kurt Luther, Markus Krause, Sophie Isabel Vennix, Steven P Dow, and Björn Bjorn Hartmann. 2016. Almost an Expert: The Effects of Rubrics and Expertise on Perceived Value of Crowdsourced Design Critiques. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, Vol. 27. Association for Computing Machinery (ACM), New York, NY, USA, 1005–1017. https://doi.org/10.1145/2818048.2819953