Infrastructuring for Crowdsourced Co-Design

Delcy Carolina Bonilla Oliva, István Koren, Ralf Klamma

Advanced Community Information Systems Group, Information Systems and Databases Chair, RWTH Aachen University, Ahornstrasse 55, 52056 Aachen, Germany

{bonilla, koren, klamma}@dbis.rwth-aachen.de

Abstract. User satisfaction determines the quality of a product idea. Yet it is hard to accomplish when designers are isolated from their users, creating a gap in the design practices. Co-design seeks to meet the needs of users by giving them a voice in the design process. Technology-enhanced learning provides an ideal testbed, as co-design practices on learning content are well-established between instructors, e.g. in instructional design. The challenges are first to convene geographically distributed users to collaborate on design of software applications and second to scale up to a high number of users. We present Pharos, a platform where designers can request feedback from a community of people with different backgrounds. It combines co-design with crowdsourcing to enable mass feedback. A user evaluation showed that designers preferred structured feedback from a crowd of users rather than open-ended critique from co-designers. Based on the evaluation, we discuss possible improvements of Pharos and motivate further studies. The resulting Web application is available as open source software.

Keywords: Infrastructuring, crowdsourcing, co-design, design feedback.

1 Introduction

Nowadays, Web applications form part of people's lives. Surrounded by millions of Web pages it is necessary to stand out to be reached by the user's interest. Designers are using participatory methods to collect ideas and enable users to express themselves [1]. *Participatory design* or *collaborative design* (co-design) is the inclusion of users within a development team with the goal of helping in setting design goals and planning prototypes [2]. Because users' satisfaction determines the quality of a product idea or service, they are a valuable source of ideas [3]. Users have a position of influence in the development process. Co-design seeks to meet the needs and preferences of the users for a certain service or project [4]. It not only allows users to give their opinions on predefined problems, it helps to identify the problems that need a solution.

In particular, in technology-enhanced learning (TEL) co-design practices are well-established. Instructors, teachers, learning content creators and other relevant stakeholders design courses, books, exercises and other learning objects. They receive feedback from students and pupils as well as from colleagues. In the EU project Learning Layers, co-design and participatory design techniques were at the heart of the project to design and create scaling learning applications for informal learning situations, mainly in workplace oriented learning contexts [5]. Four co-design teams were developing learning apps from different perspectives with different background knowledge and experiences. One of the main outcomes of the project was a deepened understanding that the necessary infrastructure co-evolved during the co-design process and needed to be ‘synchronized’ with them [6].

Thus, we studied the concept of ‘Infrastructuring’ in order to better understand the dynamic interplay of these processes. Star and Bowker [7] have put forward the verb ‘to infrastructure’, emphasizing the conditional, flexible and open character of the design of infrastructure process, blurring boundaries between use, tailoring, maintenance, reuse, and design. Infrastructuring includes the social environments around the project and the support of its work, therefore, tailoring and configuring the project outside professional activities. ‘Infrastructures shape and are shaped by the conventions of practice’ [7]. This means users are expected to be more engaged in the development process, give a significant investment of time rather than just answering questionnaires and give minimal input.

User satisfaction can be hard to accomplish when designers are isolated from the users, their experiences, and needs, creating a gap in the planning and design practices. This gap in the communication between the different groups of users and design team is one of the major challenges in design [8]. A possible solution for this problem is to leverage the community by giving them a voice to express or back the ideas from small groups of co-designers and filter the voice of a few to express the idea of the masses forming a system that represents or captures their needs.

Moreover, the internet has increased the availability and accessibility of the users, enabling crowdsourcing. Crowdsourcing is assembling a crowd to use their expertise [9]. Using crowdsourcing in the design process results in massive participation approaching the designers to the expectations of their users.

One of the primary goals of this project is to fill the communication gap between designers and the community. We present *Pharos*, a browser-based collaborative platform that seeks to build and maintain relationships between every person involved or related to a project identifying common goals. By gathering expectations and requirements from the crowd from the beginning of the design process, getting feedback from early designs, allowing users to redesign, showing the evolution of the project, and finally, performing tests on the resulting designs. It will create development opportunities within different circles of people with different backgrounds, by scaling the collaborative design process. Scaling is being addressed in terms of the scope of the user population. Thus, Pharos seeks the collective wisdom since “collective wisdom is often accurate” [10].

This paper is organized as follows. Section 2 describes related work in the areas of co-design, infrastructuring, crowdsourcing, and design feedback. Section 3 proceeds with a survey amongst designers that raised issues with current user participation practices, which in turn informed the requirements in Section 4. Here, we additionally detail on our conceptual design. Section 5 showcases the highlights of our implementation. Section 6 discusses the evaluation of our prototype, before Section 7 concludes this article with implications on TEL and possible future work.

2 Related Work

First, we explore the concepts of co-design, infrastructuring, crowdsourcing and design feedback. These concepts are needed for understanding how users can improve final designs and affect the structure of an application.

2.1 Co-Design

Co-design is the collaboration between trained designers and users in order to generate new ideas, concepts or improvements [8] working towards a common goal of improving their practice. Co-design is considered a category of participatory design in which all the participants, regardless of their background, have an equal say [11]. Thus, the goal of co-design is to determine what are the recommended versus undesirable approaches in terms of design [8]. Consequently, the new result will possess the combined views, input, and skills of people with different perspectives and experiences [4].

Traditional methods of design focus on the designing of 'products', where the process is organized around the product. On the other hand, new design techniques focus on designing for a purpose. This means designing to meet the needs of the people. It is no longer about the product itself. It is about experience, emotion or interaction with the product. In this manner, co-design is not only 'adjusting what should be designed' but also 'who is designing' [8].

Bradwell and Marr [4] give a complete definition of co-design taking several aspects into account:

- Participation: Co-design is a collaboration. All participants are aware of the design methodology, its inputs and outputs, its goals and current status, etc.
- Development: Co-design is a developmental process. It involves the exchange of information and expertise relating to both the subject of the design process and the process itself.
- Ownership and power: Co-design shifts power to the process empowering those in a traditional 'client' role, creating a sense of collective ownership.
- Outcomes and intent: Co-design activities are outcome-based: they possess a practical focus, with a clarity of vision and direction.
- Methodology: Successful co-design requires a methodology that supports and actively encourages its core properties.
- Environment, communication and context: A co-design project creates a safe space for input. Effective and accessible is essential to ensure a successful project.
- Checks and balances: It may be important to restrict the scope to maintain focus on desired outcomes.

Sanders and Stappers [8] recognize that users can become designers as "experts of their experiences" [12] if they are given the appropriate tools for expressing themselves. Moreover, Faste [13] considers that the creation of ideas must be consistently sustained over time throughout the design process. This means ideas should not stop flowing after an initial overflow. This task is up to the platform.

2.2 Infrastructuring

Co-design entails a shift from treating designed systems as fixed products to treating them as ongoing infrastructure, [5] broadening the focus from technology to also include its embedding community. The term infrastructure is often understood as a group of technologies, standards, tools, and applications used for a certain purpose [14]. Pipek and Wulf [15] highlight how the separation of design and use can affect the desired results since the surroundings of the product have been neglected. Information systems present the possibility of design-before-use and design in use. Furthermore, they emphasize the need to redesign and expose the features of infrastructure that need to be updated or changed, considering change as an aspect of everyday practice, and not a privilege of professional design. Karasti and Syrjänen [16] express that "if technologies are to be made useful, practitioners must effectively take up the work of design. Technological infrastructures should always be seen in relation to organized human practices, as parts of social systems".

Pipek and Wulf [15] list the different kinds of support needed for infrastructuring: basic technological support, articulation support, historicity support, decision support, demonstration support, observation support, simulation support, exploration support, explanation support, delegation support, (re-)design support.

It is necessary to distinguish between co-design concerned with design-for-use and co-design focused on design-for-future-use. The first one is a response to a known issue. The second one, also called participation as infrastructuring, can be created within a broader view. It brings together individuals to discover unknown issues, through experiences and social relations, creating a base to sustain a community and their relations [17] .

Marttila and Botero [18] highlight the need of linking active communities as part of the infrastructuring process resulting in various technical and social interdependencies. As a result of these interdependencies, it is possible to define and make visible specific features and functions required by the new system.

Communities need infrastructure to exist [19]. Communities of Practice (CoPs) have started to use, experiment with and design technologies. Thus, there is an increasing necessity to create platforms supporting distribution, exchange, consumption, production, and accessibility of ideas. Adequate communication within and between communities is considered part of the infrastructuring process due to the creation of bridges between actors and resources in different contexts and practices [18]. Infrastructuring encloses providing the means for discovering and expressing existing issues, its consequences and enroll others into the cause [17].

2.3 Crowdsourcing

Crowdsourcing has been defined as a crowd of users collaborating to build an artifact which will be beneficial to the whole community. However, Doan et al. [10] expand the term to a general-purpose problem-solving method: "It enlists a crowd of humans to help solve a problem defined by the system owners".

While distributed co-design can be challenging for geographically dispersed groups, the internet can reduce this cost and enable larger groups. Crowdsourcing has increased the availability and accessibility of the users. Crowds can perform open-ended tasks; the crowd communication can go from total isolation to open collaboration and crowd members can potentially crowdsource [20]. Leveraging the potential of the crowd could lead to a more complete panorama of the needs of the users. However, the difficulty of management of such a community will considerably increase, empathizing the need for a clear and scalable infrastructure. In the co-design and infrastructuring fields, communication between large amounts of users, especially when coming from different communities can be challenging [18] accentuating the need for appropriate tools for feedback.

2.4 Design Feedback

Managing feedback from the crowd can be challenging and most of the critique software is not designed to scale to large communities [21]. An essential concern in co-design is unacknowledged participation [22]. However, studies have shown that it is technically feasible to conduct large-scale usability studies in different crowdsourcing platforms [21, 23, 24]. Moreover, it has been found that crowd and expert critiques have comparable consistency, designers considered crowd critiques valuable. With crowd critiques designers found more issues, producing better final designs than with generic feedback [21]. Sakamoto et al. [20] have alike stated that a crowd might be more efficient and effective than an expert.

Nonetheless, producing meaningful feedback from a large number of users represents another challenge. Open-ended critique without appropriate task structuring can lead to low-quality responses lacking reasons and possible solutions.

Heintz et al. [25] have identified nine key functional requirements for enabling users and developers to manage feedback and have divided them into user requirements and developer requirements, these are presented in Table 1.

Table 1. Key functional requirements for enabling user and developers to manage design feedback [25]

	Requirements	Description
User	Interactivity	Work with interactive online prototypes
	Annotation	User can put textual feedback to a specific element of the design
	Creativity	Support drawing to provide graphical feedback
	Collaboration	Users can provide annotations collaboratively
	Access	The tool can be access with Internet connection and without installation
	Instructions	Instructions are provided to give user support
Developer	Activity	User activity data are collected in addition to feedback
	Aggregation	Different user data aggregation

Co-design at large scales could provide unwanted results when the input of the few has the potential to dictate a design for the many [4]. Thus, the design of an appropriate interface, with satisfactory usability, representation, and visualization techniques can prevent mass confusion and improve co-design at a community level [26].

2.5 State of the Art

Currently, several platforms that offer co-design or crowdsourcing capabilities exist. (1) InVision [27] is a collaboration platform for designing interactive prototypes with a small team of designers. (2) Gallery [28] It is a free collaborative tool for uploading designs, gathering feedback and tracking versions of a design with no rapid prototyping. (3) DisCo [11] enables storylines, annotations, and comments within the tool, however, it does not allow prototyping and collaboration is meant to be done in small teams. (4) Pdot (Participatory Design Online Tool) [25] provides comments, annotations, and sketches. However, the users are not able to provide their own design versions. With (6) CrowdCrit [21] designers are able to receive structured feedback on early designs from a non-expert crowd but participants are not able to re-design the initial proposal.

Most of these platforms are not designed to handle a large number of critiques. Whereas, applications that indeed are designed for crowdsourcing do not typically allow users to create or modify prototypes. We have identified a gap in the current state of the art for a platform that facilitates infrastructuring, co-design activities and crowdsourcing. This platform would enable designers and developers to create better designs for their users, where they can listen to the user's wishes, easily manage their feedback and show them that they are being heard by rapidly adding these improvements into the prototypes.

3 Initial Survey

During the early stages of the design of Pharos an initial survey was released in order to gather insights on the current design process that developers and designers use and to collect input for their needs in their design practices.

3.1 Methodology

The initial survey seeks to identify how designers are currently designing user interfaces, if they are receiving feedback on their designs and how they tend to receive it. The research purpose of the survey is to identify if there is a need for a platform which is able to facilitate prototyping and crowdsourced design feedback. The hypothesis is that designers will prefer a single platform where they are able to create prototypes, receive feedback on their designs and apply it.

The survey consisted of twenty-two questions, divided into five different sections: (1) demographics, (2) current development technique, (3) infrastructuring, (4) user feedback and (5) opinion on the proposed platform. Nineteen questions on the survey were closed-ended to ensure that the survey would take a short amount of time to complete. Eleven of these nineteen questions used a 5-point Likert scale to allow respondents to agree or disagree with the given statements, the remaining eight questions were single or multiple choice and three questions were open-ended.

The first section collects age, gender and field data i.e. whether respondents are developers, designers or both. The second section obtains data about how they develop or design applications, for whom they design and the number of applications they create per year. The third section tries to find out if respondents are already involved in an infrastructuring process. The fourth section seeks to gather the participants' opinions and methods for collecting user feedback. Finally, the last section tries to gather the expectations on the proposed platform.

The respondents were collected by publishing the survey in different design communities (Open Source Design[1] UX Mastery Community[2], Web Designer Forum[3] and Site Point Forums[4] as well as distributing it among colleagues and personal contacts. In our research design, we followed the ethical guidelines of our university. We clearly indicated the purpose of the survey by declaring that the results are used as input for our research and tool development. Participation was voluntary, anonymous

[1] https://discourse.opensourcedesign.net
[2] https://community.uxmastery.com
[3] https://www.webdesignerforum.co.uk/search
[4] https://www.sitepoint.com/community/c/design-ux

and could be discontinued at any time during the survey steps; by submitting the online form, we obtained consent to use the data as indicated.

3.2 Results

In total, 34 responses were collected.

Demographics. All of the participants are 18 years or older, with 25 participants between the ages of 25 and 34. 25 participants are male, 6 female and 3 preferred not to say. 30 participants are developers, 15 are designers and 3 of them have management or lead positions.

Current Development Technique. On average, participants design or develop 3.1 applications per year. Around 80% of the participants do not use tools to develop applications without coding. In general, participants do not use automation in their design process.

Infrastructuring. Only 30% of the participants develop or design applications that allows users to develop new designs. Currently half of the participants already involve users in the design process. 65% do it during the entire process. 19 participants develop applications based on any other system already in use indicating that some level of infrastructuring is already present in their design process. These results describe a need for collaboration. A platform where users can request features and information can be easily shared is proposed.

User Feedback. Fig. 1 shows that 85% of the respondents consider user feedback important. From the 34 participants only two do not consider user feedback relevant. Furthermore, 70% often use the user feedback to improve their designs.

12 respondents only receive feedback from an amount of 1-5 users. Only 4 participants receive feedback from more than 20 users. However, 24 participants are interested in gathering feedback from more than 5 users.

This result highlights an appeal for crowdsourcing tools to obtain design feedback. Additionally, 66% of them use interviews, followed by the use of existing platforms and design workshops to involve users as designers. The most popular method to receive feedback is to conduct interviews, followed by surveys and A/B testing. Only 3 participants currently use crowdsourcing.

In order to obtain similar results as with interviews, commenting, annotations or sketches can be introduced. Moreover, to maintain structured mass feedback that the designers can easily assess, surveys and A/B testing can be brought into the platform.

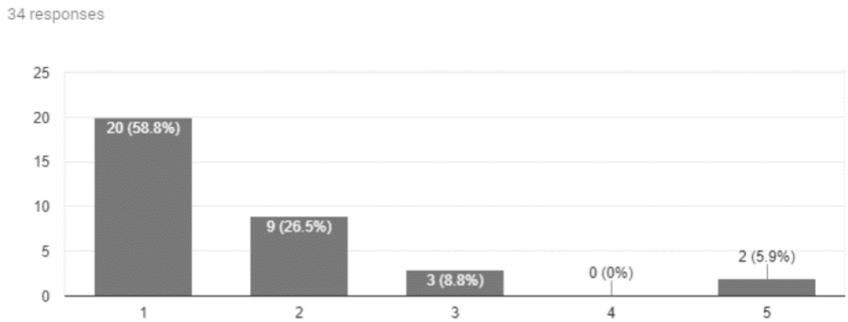

Fig. 1: Survey Result for the Statement: 'user feedback on designs is important' (1. Strongly agree - 5. Strongly disagree).

Proposed Platform. Participants show interest in gathering feedback from more than 5 users. Furthermore, 70% of the respondents are interested in a platform that allows prototyping and introducing mass feedback in the same space demonstrating that the respondents are interested in a platform where they are able to prototype and crowdsource design feedback.

When asked about the expectations of the proposed platform respondents envisioned structured, categorized and summarized feedback, live updates, logs easily readable by other programs (e.g. R), live crowdsourced-sessions, versioning and compatibility with existing tools.

The results of this survey show a desire of designers to involve the users in the design process and extract their knowledge to offer higher quality services and improve their experiences. They also show interest in scaling the design process to accommodate a larger number of users. As a solution, we suggest a crowdsourcing application to gather structured feedback. It also reveals a certain enthusiasm for a tool to improve the relationship between designers and users. In the next section the features and requirements for such a tool will be established.

4 Concept

Co-design and infrastructuring seek the knowledge and experiences of users. Pharos, the proposed platform, should enable designers and users to express themselves at different levels of creativity. In this manner, designers will be able to generate better concepts and users' expectations will be met. Moreover, infrastructuring connects participation and space, our goal is to offer means of interconnecting these two concepts. Pharos seeks to offer openness. In this manner, it is expected to achieve more than just co-design by allowing users to actually modify projects from style to features required from the projects. The main goal of this platform is to facilitate the collaborative design between designers and a large number of users with different backgrounds, (from professionals to non-experts). This means handling prototyping, collaboration and an enormous amount of feedback. Participation and infrastructuring are able to complement each other to expose and resolve issues [17]. Thus, the proposed platform will not only allow for co-design but it will also sustain infrastructuring by creating networks of various people with different roles and involving evolving social networks [29]. The platform aims to gain new insights to create more stable iterations of a project.

4.1 Requirements

Usually, ideas between partners in the co-design process in small groups are exposed with low-tech prototyping, post-it notes and art supplies [11]. However, when dealing with large scale participation, other requirements are needed. Based on the related work presented in Section 2, several requirements have been identified. They have been confirmed or added (e.g. A/B tests) by the results of the survey presented in Section 3. These requirements are, (with terms *will, should, and could* indicating priorities):

Roles. Different roles must exist to maintain a clear structure, manage the feedback adequately and provide a sense of ownership as discussed by Bradwell and Marr [4]. (1) The maintainer who seeks to obtain feedback from colleagues and users and keep them involved during the entire design process. (2) The co-designers who should be able to create new screens for the design, give open-ended feedback and share the project with voters. Finally, (3) the voter's role is to complete the design tests launched by the maintainer.

Collaboration. Pharos could enable co-design with the roles of maintainers and co-designers working together on the prototypes, annotations and discussions of ideas and conflicts emerging from the comments [4, 25].

Commenting. The maintainer and co-designers should be able to comment on the overall project, as well as on each screen, thus supporting collaboration according to Heintz et al. [25].

Annotations. Prototyping should be available for all co-designers, however, not all of them will count with time or the willingness to create new versions of a screen. Annotations enable them to give rapid feedback in a less complicated manner and thus showing their mental model through sketches. It also provides them with the possibility of adding short text over the designs, supporting collaboration, as well as creativity [25].

Voting. Voting would enable structured feedback to manage it adequately [21, 23, 24]. Voting can be done through preference and questions tests published by maintainers. Voters could either choose between two different versions of a screen or answer simple questions about a single version of the design.

Feature Requesting. Co-designers should be able to request any feature or request the removal of unnecessary traits, thus, enabling decision support [15]. Any functionality they consider necessary should be added into the discussion of the project and should be easy to follow.

Prototyping. Prototyping should be enabled for maintainers and co-designers alike. Enabling demonstration support and redesign support [15], as well as interactivity and creativity [25]. It must not be limited to people possessing coding skills. Any user without coding knowledge should be able to easily create new screens or new versions of screen by prototyping a given design and therefore allowing the target user of a project to participate in the design process.

Versioning. Versioning designs was expected from the participants on the survey presented in Section 3. Additionally, it would enable historicity support [15]. After the creation of a screen design, maintainers and co-designers could modify this design and upload it as a new version. This will enable them to see the evolution of a design, what satisfies the users and has been constant through the versions. The evolution overview or design iterations seek to point out the characteristics that are always modified, meaning that they are not usable and must be changed by the maintainer.

Design Testing. Maintainers could launch different design tests, in which they test more precise elements of the design. This is called a preference test. We expect maintainers to choose two different versions of a design and set a specific goal to get straightforward results. Maintainers can, for example, test size, type and positions of buttons, the headline, the images and the text. Maintainers should also be able to tests overall concepts from simple questions in order to gather feedback on specific features or elements. They could set up questionnaires. These types of tests are already being used by developers and designers as the survey in Section 3 shows. Additionally, they will enable the maintainers to gather the feedback, as well as collect their understating on a specific characteristic. Additionally, the preference tests and questionnaires will help maintain a structured feedback.

4.2 System Overview

Pharos aims to bring together developers and their users to create better designs. In order to achieve this collaboration environment, Pharos consists of different components. First, it uses an *authentication service* to identify users and their different roles. Namely, to identify maintainers and co-designers and grant different authorization levels and permissions to create projects and tests. Each project can consist of several screen designs. When creating a screen, the user can decide whether it would be an image or blank screen ready to be prototyped. For the second option, a prototyping tool will facilitate maintainers and co-designers to create and modify designs of a screen within a project. Each time a new prototype is created, it will be saved by using a *versioning service*. The versioning service will keep track of the iteration of a single screen. Furthermore, a new design can be created from any version. The iterations of the design will be stored on a file server. The platform will enable its users to share their designs and invite co-designers by sharing links through social media or e-mail. Aside from the creation of new versions, the feedback management consists of three main parts. These are *threaded comments*, *annotations* and *votes*. Moreover, a *testing service* will be available. With this service, maintainers will be able to launch preference tests with two different iterations of the design, as well as, questions tests, where they can create a small questionnaire for the users.

5 Implementation

Pharos is a Web application created using Angular[5]. Several components were built to achieve the requirements explained in the previous section, namely votes, annotations, prototyping, versioning and the design test. Additionally, two widgets created as Web Components using Polymer[6] were added. All the data created from Pharos is stored as JSON objects. The data of the projects of Pharos is saved using a real-time database

[5] https://angular.io
[6] https://www.polymer-project.org

from Firebase and all the files and images are stored in GitHub[7], a well-known source code host for open source projects. It acts as a file server and a small backend was created with Node.js with the purpose of accessing its API.

Pharos' frontend and backend applications and the threaded comment service are hosted by an Nginx web server.

5.1 Services

Pharos consists of several services needed to meet all the requirements set in the previous section.

- *Authentication Service.* Pharos uses Learning Layers Login[8] as a single sign-on OpenID Connect authentication server, which is used for a number of learning services. This means Pharos delegates the authentication process to Learning Layers and does not handle identity management. Credentials verification and credentials storage are managed by Learning Layers.
- *Request Service.* Requirements Bazaar[9] is used as the request service. It is an open source platform to gather requirements for projects. Users and designers can share ideas, requirements and problems about a project. It can be used during the entire design process. Designers are able to describe initial ideas and get early expectations from their users.
- *Comment Service.* The las2peer Threaded Comment Service[10] offers comment functionality. It is possible to create a comment thread, set up permissions and post a comment to this thread using a REST API.
- *Prototyping Tool.* GrapesJS[11] is used as a prototyping tool. GrapesJS is an open-source Web Builder Framework used to build HTML pages that can be effortlessly integrated with Angular 5. After the maintainers and co-designers finish prototyping, they can save their designs as HTML and CSS code. The results are saved by the versioning service and used for capturing the preview image of the respective screen.
- *Versioning Service.* GitHub[12] is used as a file server to store the different iterations of a design. GitHub enables the versioning of a design through the commits which record the changes of the repository. In this platform each commit represents the change of a single file. All the changes of a screen are saved as new commits with GitHub API. New version can be created from any version, thus, users can easily aggregate to others co-designers' ideas, and the maintainer is able to see which versions are preferred by the co-designers and what direction should be pursued.
- *Voting Service.* The voting service enables approval voting, where users can upvote or downvote an element. The votes are stored in Firebase as a property of each object and create a small Angular component receives the number of upvotes and downvotes.
- *Annotation and Sketch Service.* Annotations are created on a canvas containing the image of the current selected screen version. Users are able to

[7] https://developer.github.com/v3/?
[8] https://api.learning-layers.eu/o/oauth2
[9] https://requirements-bazaar.org
[10] https://github.com/rwth-acis/las2peer-ThreadedComment-Service
[11] http://grapesjs.com
[12] https://github.com

navigate and examine them through the list of annotations. Finally, once the co-designer decides to save the annotation an image of the canvas will be created and saved using the versioning service.

- *Testing Service*. In order to receive structured feedback from a crowd of users a testing service was created within the Angular application. It consists of three main parts: the test administration panel (shown in
- Fig. 2), the preference test and the question test. The test administration panel allows the maintainers to easily create tests which are then saved in Firebase. It collects the data of a design iteration using the versioning service and creates and image of this version. The test files are also saved using the versioning service. Finally, after creating a URL is shown to the maintainer. The maintainer can then share the URL with the crowd to receive Feedback. The test results are also shown in the administration panel. The tests themselves are shown to the voters in a separate component within the Angular application, where they do not have access to the rest of the feedback created by the co-designers. They can easily participate on the test and submit their responses.

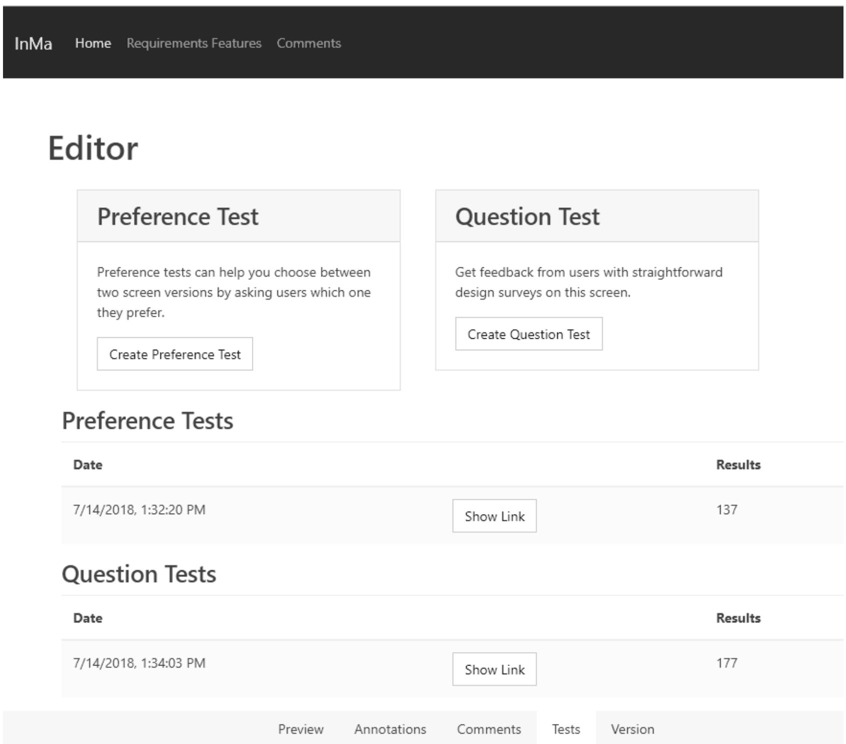

Fig. 2: Test Administration Panel

6 Evaluation

To evaluate the platform, we correlate its features against the different requirements gathered in related work. The user evaluation was done by recruiting users in the different roles: maintainers, co-designers and voters, as well as, gathering the maintainer's thoughts on the feedback received with the platform. Finally, in order to gather the usability perceived by the users, The System Usability Scale (SUS) [30] was used with all the users covering the perception of the maintainers, the co-designers and voters.

6.1 Functional Completeness

Pharos achieves all the requirements set in Section 4. Most of these requirements originated from the related work presented in Section 2. Thus, to evaluate the design process of our platform, the requirements by Bradwell and Marr [4], Pipek [15] and Heintz et al. [25], are analyzed.

The co-design process in Pharos is evaluated with the definition given by Bradwell and Marr [4]:

- Participation: Pharos enables collaboration. The co-designers are invited by the maintainer making them aware of the design methodology, its inputs and outputs, and the goal of the project.
- Development: Pharos enables the exchange of information around a project and in this manner improves the design process itself.
- Ownership and power: a balance between the co-designers and the maintainers is created by granting the co-designers part of the power in the design process. Thus, creating a sense of collective ownership.
- Outcomes and intent: the intent of the co-design process can be clearly stated by the maintainer when creating the project and the initial prototype.
- Methodology: with the versioning and prototyping of the designs, the creative intent is not only kept but encouraged.
- Environment, communication and context: Pharos creates an accessible space where input and communication between participants is effective and flexible.
- Checks and balances: A balance of the received critique is achieved by the design tests which give structured feedback to the maintainers.

Pharos creates a co-design environment that is able to expand the spectrum of participation, going beyond answering simple questionnaires and getting minimal user input to a significant ownership from part of the users or co-designers. It seeks to go beyond enabling participation and co-design towards enabling infrastructuring. Pipek [15] states the eleven different types of support needed for infrastructuring. These types of support are examined to evaluate the platform. The comment functionality enables (1) articulation support and can also be used to enable (2) basic technological support. The versioning service facilitates (3) historicity support. The design tests provide the means to enable (4) decision support, (5) demonstration support and (6) observation support. (7) Simulation support and (8) (re-) design support are implemented by the prototyping tool. The screen view on the Web application of Pharos enables (9) exploration support and (10) explanation support. Finally, (11) delegation support is facilitated by the different roles in Pharos. Therefore, Pharos fulfills all the types of support needed for infrastructuring enabling developmental and appropriation activities.

Furthermore, we evaluate Pharos against the key functional requirements stated by Heintz et al. [25] for enabling users and developers to manage feedback. These requirements were presented in Table 1. The users of Pharos are invited to collaborate on a design project by the maintainer. This enables them to work with interactive online prototypes from any designs made by other participants. They can add textual feedback as well as sketches or annotations. In other words, the user requirements are met. The developer requirements of activity and aggregation are met, since the maintainers can see the activity of the users by looking into the feedback and the crowdsourced test results are clearly presented. The resulting designs made with the prototyping tool can be exported into HTML and CSS code. However, the last developer requirement is not accomplished since the resulting data is not exportable to be used for statistical software.

6.2 User Evaluation

The user evaluation gave two developers the opportunity to give an initial design and share it with co-designers and with a crowd of voters. The user evaluation was divided into three parts according to the role of the users. First, the maintainers could set the initial project. Second, the co-designers could give feedback and co-design the project. Finally, feedback was collected from a crowd of users using Amazon Mechanical Turk[13].

Setup

In total 121 users participated in the user evaluation. Two users had the role of maintainers. Nine users participated as co-designers. The crowd of voters was composed of 110 users. All of the users worked on the same sample project. They were asked to co-design an infographics maker called InMa. Infographics are graphic visual representations of information, data or knowledge intended to present information quickly and clearly. The goal of the user evaluation was to co-design this tool.

The user evaluation begins with the maintainers. The maintainers were asked to participate in two different parts of the user evaluation. First, with the creation of the project, an initial design and the design tests. The second part consisted of reviewing the feedback. A crowd of non-expert users were given the design tests created by the maintainers. The goal of this part of the user evaluation was to gather structured feedback for the maintainers using Pharos. Finally, all different types of roles answered a System Usability Scale (SUS). The SUS is a questionnaire used for measuring the perceived ease of use of a given system. It consists of 10 questions that can be answered with a 5-point Likert Scale (1 Strongly Disagree - 5 Strongly Agree). Within industry standards the average SUS score is 68. A score higher than 68 can translated as good rate of a system. A score higher than 80.3 represents an excellent score [30].

Results

Maintainers. The maintainers had the task to create the project and set the initial flow of the co-design process by creating a sample prototypes and design tests. Two users participated with the role of maintainers. They are both developers and do not consider themselves designers. They were able to create the sample project, create designs of the welcome page and the editor page and design tests for a crowd of voters using given designs. The SUS score of the maintainers was 77.5.

[13] https://www.mturk.com

When presented with the feedback of the co-designers and voters, they found the feedback from the voters more helpful as the feedback from the co-designers. This was due to the ease provided by the structured feedback, while the feedback from the co-designers required more time to analyze.

Co-designers. The goal of the co-designers on this user evaluation was to create feedback of the sample project. Nine users participated as co-designers. Eight are developers and one of them is also a designer and one has a management position. The average SUS score is 66.9 of the co-designers and the responses are shown in Fig. 3. In general, they are neutral on using Pharos frequently. They did not find the platform to be complex. They showed a positive tendency regarding the ease of use of the platform. The co-designers, as well as the maintainers, found the system well-integrated, consistent and easy to learn. Half of the respondents declared that they likely would reuse Pharos. However, they expressed discontent with the prototyping tool. Moreover, as co-designers, they did not have enough motivation to give feedback about the design. Co-design and infrastructuring activities need a certain level of engagement from part of the participants. This was not present for most of the co-designers. We believe this situation led to less meaningful critique.

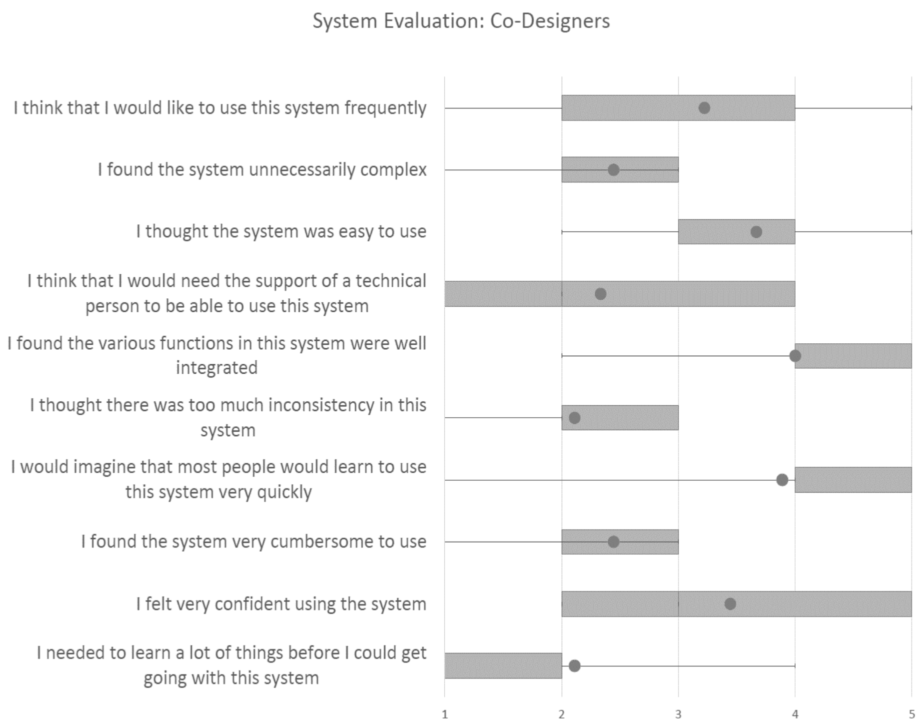

Fig. 3: SUS Results for Co-Designers (1 Strongly Disagree - 5 Strongly Agree)

Voters. The voters had the task to complete the design tests created by the maintainers. They were paid US$0.30 for performing the survey. When setting the task on the crowdsourcing platform, no approval rate was given. This led to many low-quality responses. Many of the participants that completed the SUS questionnaire did not participate in either of the design tests available. Using the data that Amazon provides, it was possible to reject responses. The rejected participants spent less than 120 seconds doing the task. That was less than half of the average time. 210 results were received, however, only 110 were accepted. From the accepted responses, 36 respondents are developers, 39 are designers and 35 participants work on other areas.

The voters were given a survey with two links for the different design tests. On the first test, they had to select between two designs to choose their preferred location of the tools panel of the sample project. On the second test, they had to answer two questions about the given design. The average SUS score is 78.5. In general, voters would like to use the system frequently. They did not find the platform to be complex. They found Pharos is easy to use, well integrated and consistent. They do not need the assistance of a technical person to participate on the design tests. Participating on the design test does not have a steep learning curve and it is not complex. In general, they did not think it was cumbersome to use and they felt very confident using Pharos.

In general, all types of users found Pharos to be consistent and well-integrated. The average SUS score of all types of users is 77.7, meaning that overall, Pharos provided a satisfactory experience. The voters showed the highest results among the three types of users resulting in meaningful crowdsourced feedback. On the other hand, the co-designers scored the lowest. Most of the concerns were directed towards the prototyping tool. We believe that the co-designers did not have a real interest in the design they were working on, this led to less meaningful feedback from their part. Thus, the maintainers preferred the structured feedback from the crowd.

7 Conclusion and Future Work

In this paper we presented Pharos, a Web application that enables infrastructuring for crowdsourced co-design. With an initial idea survey, we identified a need for a platform that facilitates infrastructuring, co-design activities and crowdsourcing enabling designers and developers to create better designs for their users. With the results of this survey and the research gathered in the related work we identified several requirements for a platform that improves the relationship between designers and users. The requirements seek to facilitate the task of involving users in the design process and extract their knowledge to offer higher quality services and improve their experiences. As well as fulfilling the need for a crowdsourcing application to gather structured feedback.

We proposed and developed Pharos. It offers a way to fill the communication gap between designers and the community. Pharos enables designers to submit designs and request feedback from other co-designers as well as from an entire community of people with different backgrounds. This leads the designers to implement better designs and increase user satisfaction.

The platform was evaluated by many users in different roles. The user evaluation showed that the users found Pharos to be consistent and well-integrated. In general, Pharos provided a satisfactory experience. The maintainers preferred the received feedback coming from the crowd because of the structured design test results. The open-ended feedback from a smaller group of co-designers required more time and effort to be reviewed. Additionally, we believe that the co-designers did not have a real interest in the design they were working on, this led to less meaningful feedback from their part. The responses from the crowd also produced irrelevant results, however this was easier to manage thanks to the crowdsourcing tool used.

While Pharos fulfills all the types of support needed for infrastructuring [15], infrastructuring is not yet well integrated. Participants must go beyond giving feedback towards shifting the trajectory of a project. Pharos aims to offer means to build and maintain relationships between every person involved or related to a project identifying common goals.

7.1 Future Work

While Pharos fulfilled the requirements, the user evaluation highlighted multiple possible improvements for the platform. Co-designers expressed the need of a more extensive annotations tool, where they are able to create annotations with different colors, pencil styles, thickness and also be able to erase them. For a higher usability, voters should be able to change their votes. Furthermore, the view of the versioning service can be improved into a tree view. This would allow maintainers to improve how they follow the history of changes of a design. The greatest improvement must be achieved with the prototyping tool. Maintainers and co-designers expressed discontent while creating new designs. It was not as intuitive as they expected.

As mentioned in Section 2, unacknowledged participation is major concern. Remote collaboration by itself can enable an increase in the co-design practices, giving a higher relevance to this issue. Natural language processing approaches like text summarization, keyword extraction, or topic modelling, could be used to structure mass feedback at the co-design stage. Moreover, presenting prototypes and annotations from a crowd in a meaningful manner is a challenge and it would entail further research. These approaches may help in recognizing participation; however, further research is also needed to ensure representation of minorities. In the context of technology-enhanced learning, a possible solution is to crowdsource within the target community by providing access to a wider group of instructors, e.g. in the context of open educational resources (OER).

In order to recognize participation and provide a representative crowd, encouragement is needed [24]. While some sites try to motivate users to take part in the experiments by offering money or rewards, some others have different kinds of incentives e.g. games offer satisfaction as incentive. More importantly, crowds will perform tasks they find interesting, thus making the motivation incentive entertainment [20]. On the other hand, the motivation of the co-designers must come from the expectations of the designed system. This means, co-designers must possess a shared interest in result of the platform, in this manner they will be able to express their goals and ideas. When the crowds are limited to an existing community, e.g. a CoP, the community members strive to be better, being this the motivation itself.

We believe that in order to obtain better results, new studies should not only arouse real interest in the subject among stakeholders, but they should be done through the course of several months, giving the participants the opportunity to invest time and effort into the project. In this manner, users could feel responsible for the project which should result in better feedback. We envision to directly embed our findings and the capabilities of Pharos in various applications like learning management systems and MOOC platforms. A formative real-world evaluation could contribute to better understand co-design practices in the learning domain. Here, further aspects beyond the visual design like the creation of learning material could be crowdsourced with little adaptations of our prototype. Finally, we foresee many possible use cases beyond learning, including mobile and immersive applications.

Acknowledgments. The research leading to these results has received funding from the European Research Council under the Seventh Framework Programme project “Learning Layers” (grant no. 318209) and the European Union’s Horizon 2020 Programme through the project “WEKIT” (grant no. 687669). Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2023 Internet of Production – 390621612.

References

1. Steen, M., Manschot, M., Koning, N. de: Benefits of Co-design in Service Design Projects. International Journal of Design 5, 53–60 (2011)
2. Carroll, J.M., Chin, G., Rosson, M.B., Neale, D.C.: The Development of Cooperation. In: Boyarski, D., Kellogg, W.A. (eds.) Proceedings of the conference on Designing interactive systems processes, practices, methods, and techniques - DIS '00, pp. 239–251. ACM Press, New York, New York, USA (2000)
3. Kristensson, P., Magnusson, P.R., Matthing, J.: Users as a Hidden Resource for Creativity. Findings from an Experimental Study on User Involvement. Creativity and Innovation Management 11, 55–61 (2002)
4. Bradwell, P. and Marr, S.: Making the Most of Collaboration. An International Survey of Public Service Co-Design. London, http://lx.iriss.org.uk/content/making-most-collaboration-international-survey-public-service-co-design-demos-report-23 (2008)
5. Star, S.L., Ruhleder, K.: Steps Toward an Ecology of Infrastructure: Complex Problems in Design and Access for Large-scale Collaborative Systems. In: Smith, J.B., Smith, F.D., Malone, T.W. (eds.) Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94), pp. 253–264. ACM; ACM Press, New York, New York, USA (1994)
6. Renzel, D., Koren, I., Klamma, R., Jarke, M.: Preparing Research Projects for Sustainable Software Engineering in Society. In: Proceedings 2017 IEEE/ACM 39th IEEE International Conference on Software Engineering (ICSE) (2017)
7. Star, S.L., Bowker, G.C.: How to Infrastructure. In: Lievrouw, L.A., Livingstone, S. (eds.) Handbook of New Media: Social Shaping and Consequences of ICTs, pp. 151–162. SAGE Publications, Ltd, 1 Oliver's Yard, 55 City Road London EC1Y 1SP (2002)
8. Sanders, E.B.-N., Stappers, P.J.: Co-Creation and the New Landscapes of Design. CoDesign 4, 5–18 (2008)
9. Howe, J.: The Rise of Crowdsourcing. Wired magazine 6 (2006)
10. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing Systems on the World-Wide Web. Communications of the ACM 54, 86 (2011)
11. Walsh, G., Brewer, R., Joshi, A., Brown, R., Druin, A., Guha, M.L., Bonsignore, E., Foss, E., Yip, J.C., Golub, E., et al.: DisCo. In: Schelhowe, H. (ed.) Proceedings of the 11th International Conference on Interaction Design and Children - IDC '12, p. 11. ACM Press, New York, New York, USA (2012)
12. Visser, F.S., Stappers, P.J., van der Lugt, R., Sanders, E.B.-N.: Contextmapping. Experiences from Practice. CoDesign 1, 119–149 (2005)
13. Faste, H.: Opening Open Innovation. In: Deserti, A., Zurlo, F., Rizzo, F. (eds.) Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces - DPPI '11, p. 1. ACM Press, New York, New York, USA (2011)
14. Pipek, V., Syrjänen, A.-L.: Infrastructuring as Capturing In-Situ Design. In: 7th Mediterranean Conference on Information Systems (2006)
15. Pipek, V., Wulf, V.: Infrastructuring: Towards an Integrated Perspective on the Design and Use of Information Technology. Journal of the Association for Information Systems 10, 447–473 (2009)
16. Karasti, H., Syrjänen, A.-L.: Artful Infrastructuring in Two Cases of Community PD. In: Clement, A., van den Besselaar, P. (eds.) the eighth conference, p. 20 (2004)
17. Dantec, C.A. Le, DiSalvo, C.: Infrastructuring and the Formation of Publics in Participatory Design. Social Studies of Science 43, 241–264 (2013)
18. Marttila, S., Botero, A.: Infrastructuring for Cultural Commons. Comput. Supported Coop. Work 26, 97–133 (2017)
19. Engeström, Y.: Developmental Work Research. Expanding Activity Theory in Practice. Lehmanns Media, Berlin (2005)
20. Sakamoto, Y., Tanaka, Y., Yu, L., Nickerson, J., Nickerson, J.V.: The Crowdsourcing Design Space. In: FAC'11 Proceedings of the 6th international conference on Foundations of augmented cognition: directing the future of adaptive systems, 6780, pp. 346–355 (2011)
21. Luther, K., Tolentino, J.-L., Wu, W., Pavel, A., Bailey, B.P., Agrawala, M., Hartmann, B., Dow, S.P.: Structuring, Aggregating, and Evaluating Crowdsourced Design Critique. In: Cosley, D., Forte, A., Ciolfi, L., McDonald, D. (eds.) Proceedings of the 18th ACM

Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15, pp. 473–485. ACM Press, New York, New York, USA (2015)
22. Lindley, S.E., Thieme, A., Taylor, Alex S., Vlachokyriakos, Vasilis, Regan, T., Sweeney, D., Taylor, A.S., Vlachokyriakos, V.: Surfacing Small Worlds through Data-In-Place. Computer Supported Cooperative Work (CSCW) 26, 135–163 (2017)
23. Vliegendhart, R., Dolstra, E., Pouwelse, J.: Crowdsourced User Interface Testing for Multimedia Applications. In: Chen, K.-T., Chu, W.-T., Larson, M., Ooi, W.T. (eds.) Proceedings of the ACM multimedia 2012 workshop on Crowdsourcing for multimedia - CrowdMM '12, p. 21. ACM Press, New York, New York, USA (2012)
24. Xu, A., Huang, S.-W., Bailey, B.: Voyant: Generating Structured Feedback on Visual Designs Using a Crowd of Non-Experts. In: Fussell, S., Lutters, W., Morris, M.R., Reddy, M. (eds.) Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14). CSCW'14, pp. 1433–1444. ACM; ACM Press, New York, NY, USA (2014)
25. Heintz, M.M., Law, E.L.-C., Govaerts, S., Holzer, A., Gillet, D.: Pdot. In: Jones, M., Palanque, P., Schmidt, A., Grossman, T. (eds.) the extended abstracts of the 32nd annual ACM conference, pp. 2581–2586 (2014)
26. Piller, F.T., Schubert, P., Koch, M., Möslein, K.: Overcoming Mass Confusion. Collaborative Customer Co-Design in Online Communities. Journal of Computer-Mediated Communication 10, 0 (2005)
27. InVision: InVision. Digital Product Design, Workflow and Collaboration, https://www.invisionapp.com/
28. Google: Gallery. Material Design, https://material.io/gallery/
29. Iversen, O.S., Dindler, C.: Sustaining Participatory Design Initiatives. CoDesign 10, 153–170 (2014)
30. Lewis, J.R., Sauro, J.: The Factor Structure of the System Usability Scale. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B. et al. (eds.) Human Centered Design, 5619, pp. 94–103. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)