

User-centered development of social collaboration software

Raja Gumienny, Lutz Gericke, Markus Dreseler, Sebastian Meyer, Christoph Meinel

Hasso Plattner Institute Potsdam

Prof. Dr. Helmert Str. 2-3, Potsdam, Germany

Email: {firstname.lastname}@hpi.uni-potsdam.de

Abstract—Social networking functions make their way through more and more applications for private and professional use as they encourage participation and content contribution. However, designing and implementing components that attract users and truly support content creation is not a trivial task. In this paper, we present our user-centered approach of integrating social network components into a collaboration software system. The Tele-Board system is designed for supporting creative teams in their synchronous and asynchronous whiteboard work via distances. Especially for creating awareness on who did what when and documenting the project progress in general, special functions were needed. We present our procedure and challenges as well as the implementation and user feedback on the new components.

Index Terms—user-centered, social networks, collaboration, whiteboard, asynchronous

I. INTRODUCTION

Collaboration among regionally and timely separated teams is challenging due to several difficulties. One of them is the problem that team members do not know who worked at the project at what time and for how long, who else was involved, and teams have difficulties to communicate about their work in general. To some extent, these problems could be overcome by documentation and information on the project progress. In practice, however, documentation is rarely done because this implies extra work for all team members who are working in a tight time schedule anyhow. Due to time restrictions, team members often focus on their own work and communication in general is neglected. In order to encourage communication, concepts from social computing could be beneficial [14].

In our system, we want to overcome this burden and find an easy and fast way to document collaborative work where as much as possible is done automatically, but still the needs of all team members are fulfilled. For their collaborative work, people are using our “Tele-Board” system [10], which is designed for collaborative creative work for geographically distributed teams (see chapter II for more details on the system). We already had some ideas on documentation but we were not sure if the things we were planning were really important to the teams who were actually using the system. Additionally, we did not know which elements were most important to the users and including everything would have overloaded the user interface.

Therefore, we chose a user-centered development approach

in order to identify the most important functions and to find a way of documenting that would support people in actually doing the documentation as effortlessly as possible. In this paper, we present the chosen approach and show how it helped us to gather user requirements and how to include them in the implementation. Especially for systems or functions that focus on social aspects, we considered the user-centered approach being very helpful as we could directly find out whether the social behavior we intended was addressed or not. Furthermore, we will demonstrate our solution of integrating a news feed to the Tele-Board web portal and show the related implementation details.

II. SOCIAL COMPUTING FOR TELE-BOARD

Tele-Board is designed for collaborative creative work over distances. The system especially supports teams who are applying methods like *Design Thinking* [2], where people write at whiteboards and use a lot of sticky notes. As conventional whiteboards and paper sticky notes are difficult to share via distances, we developed a software suite as a digital equivalent of the traditional tools.

The Tele-Board system consists of different components: a *web portal*, a *whiteboard client*, *sticky note pad applications*, and a *server component*. The *web application* is the entry point and main administration interface for design thinking *projects* and whiteboard *panels*. A project is a collection of several panels and has different members who are allowed to work on the respective panels. A panel is the direct equivalent of a traditional whiteboard, i.e. in the beginning it is a blank surface that can be filled with drawings, handwritten text and sticky notes. In order to work on a panel, users can start the panel from the web portal and the Tele-Board *whiteboard client* will open. The whiteboard client is a Java application and thus runs on any computer. Ideally, the computer is connected to digital whiteboard hardware, such as a SMARTBoard¹, which supports finger- and pen input and this way provides a “whiteboard feeling”. As an equivalent to paper sticky note pads we created different applications for writing sticky notes. There is another Java application dedicated for Tablet PCs and pen input. For finger input it is possible to use the corresponding Apps on an iPad, iPhone or Android device. The *server component* coordinates all communication streams

¹<http://www.smarttech.com/>

and saves all whiteboard interactions to a database. This way, it is possible to view the *history* of each panel in the web portal (see figure 1). As all events are stored, users can go back to any point of the whiteboard content development and see what has happened at the board. Especially for team members who could not be at a shared whiteboard meeting this function helps to understand what their colleagues have been doing (for more information on the Tele-Board system and the history functionalities see our prior work [9], [10]).

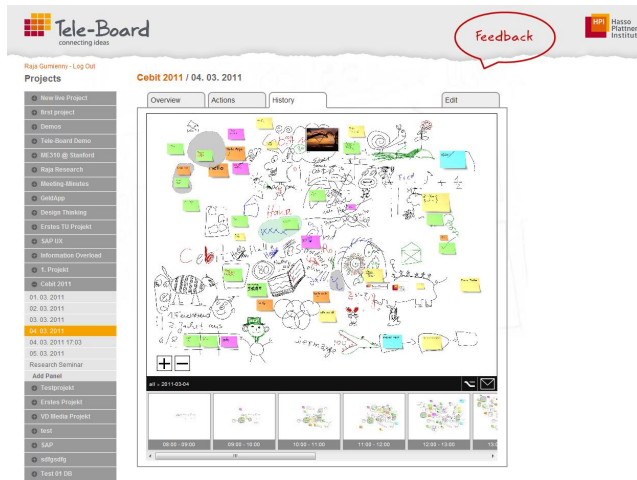


Fig. 1. Screenshot of the Tele-Board web portal and its history view

Admittedly, this history functionality does not solve the problem that users do not know when interesting work occurred. Sometimes you do not know that colleagues have been working at the whiteboard, particularly if you are located in different countries or even continents. In stressful times, people may forget to have a look at the history page and if you are member of several projects (e.g. managers) it is time-consuming to have a look at all panels to see if something has changed.

Of course, there are other possibilities to inform team members about work sessions and related tasks. But – as already stated before – documenting the own work is often neglected in stressful project work. Therefore, we were searching for a way of documenting that is easy and fast to accomplish and even encourages people to do so. Based on their popularity and high participation rates, social network functions were a promising option for our requirements. However, there are various functions and probably only some of them would fit to the project and panel structure of the web portal. How to implement and adapt them to our users' needs was challenging.

Therefore, we chose a *user-centered design approach* for the development of these new portal functions. In literature and practice, user-centered design can be understood in a variety of different ways, we consider it similar to Vredenburg et al. [13] as “the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multi-disciplinary approach”. Within this approach, a variety of different methods may be involved [13], [15], whereas we

consider the most important ones to be: user research (aka design research or user requirements analysis), prototyping and testing.

In section IV we show how we carried out these methods with regard to the “documenting whiteboard work” challenge. In the following section, we will briefly give an overview on similar documentation and social computing functions as well as comparable user-centered design approaches to web development.

III. RELATED WORK

Already in times before social networking sites became popular, annotations and notifications were used in order to create awareness among team members working on the same projects and thus facilitating asynchronous communication. Brush et al. [3] created a system with enhanced notifications for making people aware of annotations made in web discussions. The authors derived their design considerations from literature research and usage analyses of existing systems. Finally, they did a field study and could prove that the awareness improved with their notification enhancements. Another case study on how web annotations in Microsoft Office documents are used for asynchronous collaboration is done by Cadiz et al. [5].

Since social networking sites such as facebook, MySpace or LinkedIn grew successful in private and professional contexts, a lot of research focused on their usage and participation. Burke et al. [4] for example, did an analysis on feed contribution of newcomers in social networking sites and found out that active participation of their contacts stimulates newcomers to add content themselves. Other research, e.g. [12], proves that pleasure and fun of use are important factors to encourage people to contribute to a network. Participants like to have a look what has happened and rather do not have a special purpose while visiting these sites.

Other research focused on social networks in enterprises [6], [7] where they analyzed profile pages, connections and which content users usually share. The authors found that the most common pieces of information are comments on activities, followed by contacts and photos. They also analyzed how incentive mechanisms would improve participation, and besides pointing systems, the contribution of their contacts is a very important factor for participation [7].

Most of the aforementioned research used some user-centered design methods such as interviews, observations or testings. Beacker et al. [1] explicitly present their user-centered approach, similar to our research in this paper. They conducted several interviews, created two prototypes and reported on the testing of their system for collaborative writing software. This way, they could prove that the user-centered approach is very useful for creating collaborative software that will be used. Especially for encouraging participation, social technologies present opportunities, but their design and implementation is also challenging. Therefore, an iterative design process is vital [11].

As outlined here, there is research on annotations and notifications for asynchronous collaborative work and there

is also research on the effectiveness of social networks in enterprises. However, combining annotation mechanisms with social network aspects for improving collaboration is still not in major focus yet. In this paper we present our findings on how to combine the best of both worlds with the help of a user-centered approach.

IV. METHOD AND PROTOTYPING

In the beginning of the project, we wanted to find out, where problems of the users are, in order to identify the functional requirements. It was important to deeply understand and evaluate the context our users were working in. Of course, it was most valuable to study people within our target group, i.e. with a Design Thinking background and examine how they interact with the portal and the prototypes. Additionally, we included participants with different scientific backgrounds, in order to uncover a variety of usability problems.

We asked two groups of users to participate in the study: students from the *School of Design Thinking*² in Potsdam, Germany and students of the *ME310*³ course. At the School of Design Thinking, students from different scientific disciplines learn methods of creative problem solving. They are located at one location and use traditional whiteboards and sticky notes for their work. None of the participants we interviewed had used the Tele-Board web portal before.

Within the ME310 course (Mechanical Engineering course at Stanford University) teams are usually distributed over the world, e.g. at the Hasso Plattner Institute in Germany and at Stanford University in California. Except for some kickoff meetings, these teams are always physically separated. For about nine months they work on a task, which has been given by a corporate liaison, using the Design Thinking methodology. At the time of user research, these students had already tried out the Tele-Board system and thus were partly used to the state of the web portal at that time.

A. User Research

To understand the people's perspective in a Design Thinking context, interviewing was the preferred method during the early phases of user research. These interviews were captured on video. Furthermore, the interviews were conducted by two interviewers so that one person could focus on the test person and the second would write down the observations immediately. With the help of these notes, it was easier to discover interrelations and hidden needs within the video recordings. Open questions were used to encourage people to tell freely about their experiences. They told us about their work environment and activities. The questions became more specific towards problems with their work with whiteboards and - if applicable - with the Tele-Board web portal. A brief introduction to the idea of a "social Tele-Board" concluded the interview, to let people elaborate about their thoughts and wishes for such a solution. Because of the instruction to think

aloud, many hints and ideas for the portal and Tele-Board in general could be captured.

1) *Users' Problems and Ideas*: There were basically three major problem domains emerging from the interviews:

Especially in the School of Design Thinking, it is not uncommon that single team members are missing at particular days. When these team members join the team again they have to be updated by their colleagues. Often, the gaps of knowledge will be filled face-to-face, whereas it is difficult to explain complex whiteboard content without seeing its development process. It is also a problem for the teams to explain their ideas and concepts when the whiteboards are already cleaned. Although participants often take photos of the boards, it is not very common to look at these pictures again because of image quality and missing explanations.

The interviewed ME310 teams have a more general issue, which can be reduced to the problem of distribution. In their daily routine, joint sessions are hard to organize, especially due to the time shift (9 hours between Germany and California) between the locations. Documentation of each step of their work becomes extremely important in order to increase traceability. Participants mentioned that it would be nice to see an overview of all recent activities at the whiteboards.

They also stated that it is typically hard to understand whiteboard content for people not being present during the creation of it. To make the content and the connected thoughts tangible, explanation of others is needed. Therefore, it is necessary to capture who was involved in each session. The ME310 students also noted the problem that it is hard to convince others (e.g. teachers) of ideas and complex concepts with the help of a whiteboard, because the content is only "living" with people's explanations. Test persons mentioned that it would be beneficial to have a chronological whiteboard history, which is constantly documenting the creation process.

Users also demanded for additional meta information, as for example the status of the content or comments of the partner team. Because there is no distinct way of storing meta data for a Tele-Board panel, they managed with - what they called "meta sticky notes" - i.e. using sticky notes to "tag" other stickies. However, this mix of regular content and meta data was not clearly distinguishable afterwards.

The School of Design Thinking students also mentioned the need of a time-lapse playback of all changes of a single whiteboard. Even more helpful would be time-lapse videos with additional audio comments. Beyond that, a public hyperlink to open the panel in a read-only view, which can be shared with colleagues and friends, was requested.

2) *Insights and Conclusion*: User research revealed that users are requesting an extension of the web portal. People see a demand for adding more context information to the whiteboard sessions. These information needs were ranked by the participants as follows:

- *What* has changed?
- *Who* was active?
- *When* did a change happen?

²http://www.hpi.uni-potsdam.de/d_school/home.html

³<http://me310.stanford.edu>

We used this prioritized list as a basis for idea generation. To display all required information in an organized way, we considered to choose a timeline or a news feed structure. Though similar, we preferred the news feed because this structure does not focus on the particular time of events but rather on the content and people involved – just like the prioritization of our interviewees. Newest messages would be at the topmost position in order to be best visible for the user. The feed would be shown on the start page after logging into the web portal. That allows the team members to update themselves on the latest changes and the general project progress. Besides the automatically generated messages, user comments could also be shown here. Every portal user would have a simple view on all projects he is involved in. The feed view could also be shown on the project and panel pages. Messages would be filtered and only be shown for the specific project or panel that is selected. Recent developments of social networks have also proven that content feed structures [4] are valuable for encouraging contribution and make people feel part of a community [7], [8].

As mentioned, interviewees (of the ME310 course) liked the idea of having “Meta Sticky Notes”. This can be seen as a way of adding comments in order to express opinions or other information regarding a certain whiteboard state. Commenting could also be useful for panels in general or whole projects. We also wanted to give the possibility to comment on automatically generated messages.

Looking back at the priority list, users mentioned the individual who edited a whiteboard to be an essential information. Editing of a whiteboard will usually take place in a team of 3-6 people. However, the whiteboard client can only be started by one user of the team. From a technical point of view it is only possible to capture who opened and closed the whiteboard client. To give the user a possibility to record the information who was also present during the session, we planned to include an “after-work page” where colleagues, who were co-workers in this sessions, can be selected.

B. Prototyping

After collecting requirements from the users and having first ideas on how to realize them, we had to assess if our assumptions were right and the concepts really fulfilled the users’ needs. Conceptual flaws were most important to uncover during this early prototyping phase. Later, we developed prototypes that were more elaborated and more detailed. The better we prevented conceptual mistakes in early phases, the less modifications had to be applied after the implementation.

1) *Paper prototype*: As a first prototype we used a simple paper prototype to visualize the added functionality of the web portal. It was supposed to help us evaluating basic interaction concepts. In general, we found that paper prototypes are not a very good medium for dynamic web pages. Oftentimes, only small parts of the page change but there are several ones of them and each has to be created separately and exchanged during a test. This was confusing for the test participants. The

dynamic nature of appearing and disappearing elements is also hard to convey to users. Still, one problem could be uncovered concerning the position of a comment: it was unclear, to which entity a comment would refer to, e.g. a project or a panel.

As the creation and customization of the paper prototypes was very time-consuming and tests were not as close to reality as we expected, we realized later prototypes with a tool called *Pidoco*⁴. Existing web pages can be used and interlinked with each other. Accordingly, these clickable prototypes have a much more realistic behavior, whereas pencil-like contours are used to display the graphical elements. The sketched appearance can be crucial in order to communicate that functionality is not fixed and can easily be changed.

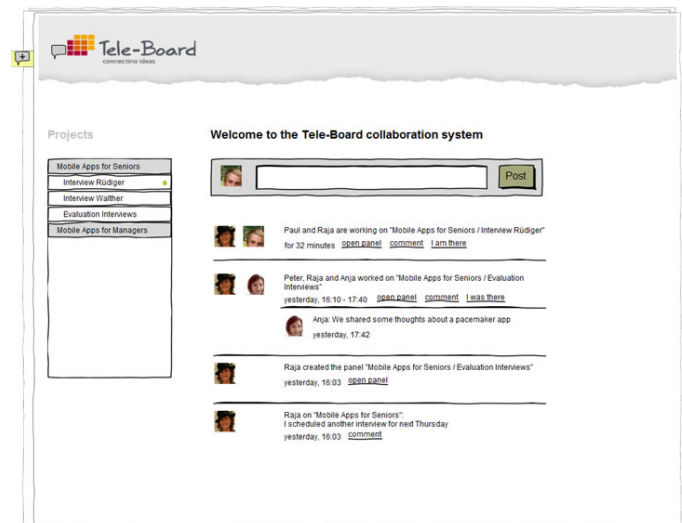


Fig. 2. First Pidoco prototype

2) *Pidoco prototype*: The Pidoco prototype (figure 2) enabled us to quickly uncover usability problems. As it is shown in figure 3, feed entries have different interaction possibilities. It turned out that it took people a long time to find the right link for a certain task they were asked to perform. The interview afterwards revealed their confusion about the different order of the links. They expected the same functionality always to be at the same position and they did not understand the difference between the entries and why there were different interaction possibilities (links).

Furthermore, the function “I am there” or “I was there” was not understood by most of the participants. These links were meant to let users express their attendance in a session after it has ended. Even with very detailed explanation, this concept could not be clarified for some participants in the study. Probably, it was difficult to understand because the participants did not actually work at a whiteboard session during the test. Their tasks were just to comment on activities or trying to open a panel. Additionally, they did not understand why they should say that they attended the session. It is a technical issue that a panel is only started by one person

⁴<https://pidoco.com/>

and does not have a relation to the real situation. As it was important for us to have an easy to understand portal page – even for first time users – we decided to turn this feature down.

A more conceptional problem was the ambiguity in which scope a message will be created on the overview page and displayed later on. Further review brought up the question if there is really the need for such messages, because every comment should have a concrete connection to the current work.

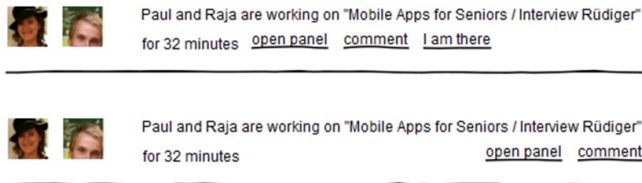


Fig. 3. Interaction possibilities of feed items: comparison of first (top) and second (bottom) Pidoco prototype

Based on the misunderstandings and usability problems that we found out with the first Pidoco prototype, we refined the second version of it. First, the “I am there” link was removed, because it often confused people and we can get the same information even more reliable out of the “after work page” (see figure 4). It turned out that it is unlikely for every user to log into the portal and use the “I am there” function.

To create a clear arrangement of the user interface elements, we decided to move every function to the same position within the feed entry and generalize the entry types (see figure 3). On the overview page, we removed the input field, because of the missing context on this page. This field is still used on the project and panel pages to post context-related status messages.

3) *Iteration on Pidoco prototype:* Users appreciated the changes we made on the prototype, so that the second test phase went even better. The workflow was perceived as more fluent by the test persons as well as the observers.

Most remarks the test persons made now, were rather related to design refinements than conceptional questions. It was mentioned that the generic text from automatic status messages (e.g. “User A worked on this panel”) should be visually distinguishable from user-created content.

Multiple participants raised the question if filling out the after work page is mandatory or optional. The necessity to fill out this page was claimed to be negative. One suggestion was to add a “skip” button to this page, to enable users to circumvent this form. Still, users perceived the after work page to be unclear. Especially the relation of the input field to the images was ambiguous. People guessed that it was possible to comment in the name of the whole team that has been selected here. Besides other deficiencies, this was changed in the next version of the Pidoco prototype (see figure 4).

After this third revision of the prototype, people’s feedback was consistently positive. Inferring from the behavior and

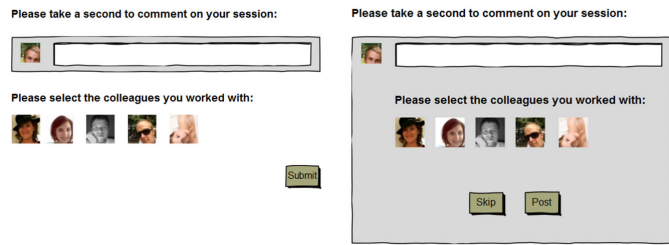


Fig. 4. After work page, first version (left) and second version (right)

comments of the test persons, they really liked the interaction concept and found it clearly understandable.

The major usability problems were uncovered by various test iterations and could be resolved on the way towards a final prototype. The result of this substantial test phase was a prototype, which is proved from a conceptional as well as from a usability perspective. This approach saved work for time-consuming changes of the final implementation. The Pidoco platform and the prototypes created with it, produce a well-suited implementation template, which can be easily implemented in a straight-forward way.

V. IMPLEMENTATION

After defining the necessary functionality through interviews and multiple iterations with different prototypes, the final implementation was realized. There are other challenges for implementing new functionality into an existing system than creating a completely new environment. Basic decisions were already made, e.g. for a web framework or the data structure in the backend. This framework can be used to implement the new functionality. The existing Tele-Board web portal is developed using the CakePHP framework⁵, which is using the MVC paradigm⁶. Especially for user centered development, it is beneficial to clearly separate the user interface from the logic behind. This means, when modifying the user interface, only small changes have to be made in the backend. This way, it is possible to quickly realize suggestions and to test the interaction.

A. Model

In figure 5 the newly added model elements are outlined. Attributes and relations to other classes are not included here. The central element is the *Feeditem*. It represents a single entry in the feed, which can be seen in figures 2 and 3. The type of a *Feeditem* can be one of: *newpanel*, *workingon*, *workedon*, *status*, *paneldelated* oder *branch*. Additionally, the time is stored at which the entry is created including the actual value - the text of the comment.

Every *Feeditem* is referencing exactly one project. During the prototyping phases, we have seen that making this relation optional is often confusing. Furthermore, a *Feeditem* can reference a panel, e.g. if a user worked on this panel and

⁵<http://www.cakephp.org/>

⁶Model-View-Controller, see <http://book.cakephp.org/view/890>

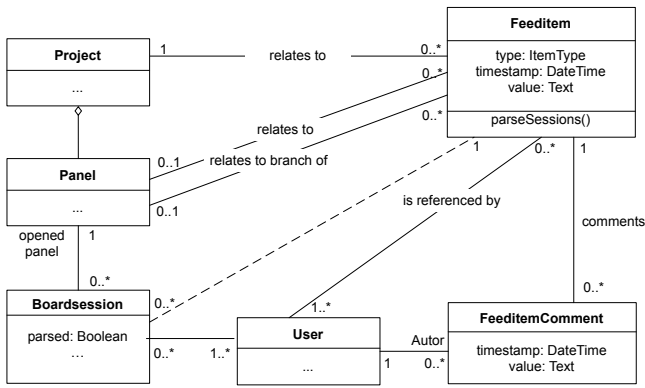


Fig. 5. Class diagram of model elements representing the Feeditem interrelations

commented on his recent session. Feeditems documenting a branch of a Panel, have a second relation to its original Panel. A Feeditem is also referencing at least one user, who executed the action (new Panel, comment on status, branch, etc.).

The method *parseSession()* creates Feeditems (workingon, workedon) based on the Boardsessions. These sessions are managed by the ServerBuddy - the server-side component handling the message flow between whiteboard clients. An additional variable *parsed* ensures that already parsed sessions are only included once in the Feeditems list. The *Feeditem-Comment* is a comment to a Feeditem.

1) *parseSession()*: This method is used to extract who worked from when until when at which Panel. This process takes several steps. First, the modification date of the last Boardsession will be compared to the last Feeditem. If there is no newer sessions, the method will break. Otherwise, it fetches all new sessions from the database. Then, we iterate Panel-wise through the sessions, differentiating between *active* and *current*, which is the currently merged overlapping session with multiple users and the next single session. In every step, it will be checked if current is within active. If so, the user of current will be added to the active list and the end time of the cumulated session will be increased to the higher value of active or current. If active is not in current, it is a new session, which will result in the creation of a new Feeditem.

All changed and closed Boardsessions will be marked as parsed and not considered for the next run, which will start when another session starts or ends.

B. Controller

The methods within the Feeditem-Controller represent the connection between the view partial⁷ and the model. The methods *index*, *status*, *comment*, and *view* should be explained a bit more detailed:

- `index(project = null, panel = null)` outputs a feed for a given context. This can be a Project, a Panel or empty, which describes the feed including all Feeditems the user is allowed to view, based on the rights

⁷partial: a view component, which can be rendered as a part of a page

for viewing Projects. This method outputs a partial, which makes them suitable for asynchronous requests to refresh the feed. This partial calls the feed method, delivering the actual view for the Panel.

- `status` and `comment` will be called asynchronously, when a user created a status text or a comment. This entry will be created and sent back as a partial to the client, which then can refresh asynchronously. This will be achieved by calling the following method.
- `view(id)` renders a single Feeditem.

Additionally, other controllers were changed, e.g. the *PanelController*, which will create new Feeditems when actions such as branching a Panel are made. A possibility to upload a profile image was added to the *UserController*.

C. View

To create a seamless integration of the new functionality in the existing portal, the social Tele-Board components had to be integrated within existing views. This refers to the Panel page as well as the project page. To reuse components, we decided for the concept of partials. The HTML-blocks can only be rendered within an existing view. Each of the mentioned views can thereby show a specific feed, without duplicating code.

To refresh a feed, we use an asynchronous mechanism, without reloading the whole page. This will be done automatically every 10 seconds and at every time a new Feeditem is created by adding a comment or changing a status message. The asynchronous functionality and dynamic effects are realized with the jQuery library⁸, which also eases the JavaScript programming for different browsers by abstracting from typical browser characteristics.

VI. TESTING

Parallel to the implementation, we did small tests on the already implemented functions. This way, we could already find out ambiguities and errors very early. In general, all test participants found the new functions easy to use. One person said that she liked our use of proven design concepts i.e. from Facebook which eases the use of new functions. Nevertheless, we also found some problems with our implementation, following below:

In the first implementation versions, the feed was displayed between other elements on the respective page, i.e. between general project and panel information. In order to make sure that the project pages still displayed all panels, we had to use an iFrame. This iFrame was a scrollable part inside the page. Users consistently rated the iFrame confusing and not usable. We could solve this problem by introducing tab views to the Tele-Board portal in general and this way separating the information on different views (see figure 6). From a technical point of view, the use of partials was also very beneficial here.

Another issue was the unclarity whether a status message was sent or not. First, we tried different animations, but then one user stated that it was confusing that the text still remained

⁸<http://jquery.com>

in the input field. After we changed this, no problems occurred anymore. Although this seems to be obvious, some easy things as this are not visible anymore if you are too much involved in implementing different functions.

In addition to the mentioned issues, users had various small design suggestions that we have included.



Fig. 6. Final version of activity feed in the Tele-Board web portal

VII. CONCLUSION AND OUTLOOK

Very often we encountered comments and remarks that were surprising or we did not think about at all. Although they seemed to be obvious afterwards, we were too much involved to see it in the beginning. We also noticed that it is important to test with realistic data as users can empathize better with the situation and feedback is more adequate.

In general, we found that the user-centered approach was very valuable for developing social networking functions as it helped us to distinguish which functions are really useful and which ones are not necessary. We got very good feedback from test participants on the current version of the implementation. Therefore, we are optimistic that these functions help them documenting their project work, and improving asynchronous communication and collaboration among the team. However, it remains to be tested if these functions really encourage contribution and content creation as we hypothesized. In the near future, Design Thinking teams will use Tele-Board for their project work and then we will investigate how much, when, why or why not they use the “Social Tele-Board” functions.

ACKNOWLEDGMENT

We would like to thank all interview partners and test participants for their valuable feedback. We also thank the

HPI-Stanford Design Thinking Research Program for funding and supporting the project.

REFERENCES

- [1] R. M. Baecker, D. Nastos, I. R. Posner, and K. L. Mawby. The user-centered iterative design of collaborative writing software. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93*, pages 399–405, New York, New York, USA, May 1993. ACM Press.
- [2] T. Brown. Design Thinking. *Harvard Business Review*, (June):84–92, 2008.
- [3] A. J. B. Brush, D. Barger, J. Grudin, and A. Gupta. Notification for shared annotation of digital documents. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*, pages 89–96, New York, New York, USA, Apr. 2002. ACM Press.
- [4] M. Burke, C. Marlow, and T. Lento. Feed me: motivating newcomer contribution in social network sites. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI '09*, pages 945–954, New York, New York, USA, Apr. 2009. ACM Press.
- [5] J. J. Cadiz, A. Gupta, and J. Grudin. Using Web annotations for asynchronous collaboration around documents. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, pages 309–318, New York, New York, USA, Dec. 2000. ACM Press.
- [6] J. DiMicco, D. R. Millen, W. Geyer, C. Dugan, B. Brownholtz, and M. Muller. Motivations for social networking at work. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work - CSCW '08*, pages 711–720, New York, New York, USA, Nov. 2008. ACM Press.
- [7] R. Farzan, J. M. DiMicco, D. R. Millen, C. Dugan, W. Geyer, and E. A. Brownholtz. Results from deploying a participation incentive mechanism within the enterprise. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, pages 563–572, New York, New York, USA, Apr. 2008. ACM Press.
- [8] J. Freyne, M. Jacovi, I. Guy, and W. Geyer. Increasing engagement through early recommender intervention. In *Proceedings of the third ACM conference on Recommender systems - RecSys '09*, pages 85–92, New York, New York, USA, Oct. 2009. ACM Press.
- [9] L. Gericke, R. Gumienny, and C. Meinel. Message capturing as a paradigm for asynchronous digital whiteboard interaction. In *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 1–10, 2010.
- [10] R. Gumienny, L. Gericke, M. Quasthoff, C. Willems, and C. Meinel. Tele-Board : Enabling Efficient Collaboration In Digital Design Spaces. In *Proc. 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD '11*, pages 47–54, 2011.
- [11] P. Hagen and T. Robertson. Dissolving boundaries. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group on Design: Open 24/7 - OZCHI '09*, pages 129–136, New York, New York, USA, Nov. 2009. ACM Press.
- [12] J. Hart, C. Ridley, F. Taher, C. Sas, and A. Dix. Exploring the facebook experience. In *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*, pages 471–474, New York, New York, USA, Oct. 2008. ACM Press.
- [13] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey. A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '02*, pages 471–478, New York, New York, USA, 2002. ACM Press.
- [14] F.-Y. Wang, K. M. Carley, D. Zeng, and W. Mao. Social Computing: From Social Informatics to Social Intelligence. *IEEE Intelligent Systems*, 22(2):79–83, Mar. 2007.
- [15] A. Williams. User-centered design, activity-centered design, and goal-directed design. In *Proceedings of the 27th ACM international conference on Design of communication, SIGDOC '09*, pages 1–8, New York, New York, USA, 2009. ACM Press.