User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM

Torsten Krohn[1], Martin Christof Kindsmüller[2], and Michael Herczeg[2]

[1] Itemis AG, Schauenburger Str. 116, 24118 Kiel, Germany
[2] Institute for Multimedia and Interactive Systems, University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck, Germany
torsten.krohn@itemis.de,
{mck,herczeg}@imis.uni-luebeck.de

Abstract. In this paper we show how a user-centered design (UCD) method can be successfully combined with an agile software development approach, namely feature-driven development (FDD), to develop the web-based information management system myPIM. This system supports users' workflow requirements in research and teaching/learning contexts. It provides bookmark, file, and reference archives, as well as possibilities for exchanging information with colleagues and students. By describing the system and its development process we show how this combination of methodologies supported our development process to create a service that truly assists the target audience and is easy to use.

Keywords: World Wide Web, online community, feature-driven development, folksonomy, information management, information sharing, internet-based collaboration, social bookmarking, social software, tagging, user-centered design.

1 Introduction

Users in research and teaching/learning contexts are highly dependent on resources that can be found in the World Wide Web. The massive growth of online information continually increases in complexity and needs both to be managed efficiently and resourcefully. Particularly, the paradigm shift from mere consumption to a new digital lifestyle of vigorous contribution of content through weblogs, wikis or the like, adds to the rapid increase of information in the WWW.

Numerous projects in the past have addressed managing large amounts of information, but with limited success. The GAB (Group Asynchronous Browsing) approach by Wittenburg et al. [1] suffers from the small overlap of the users' interests. Vistabar [2], a tool to support the users' handling with web resources, failed mainly because of the long-winded installation routine and the limited categorization options. A similar system, WebTagger [3], failed because of insufficient compatibility with many web pages. Kanawati and Maleks [4] tried to classify bookmarks automatically with multiple agents. Most people rejected the system, since this automatism did not work

M. Kurosu (Ed.): Human Centered Design, HCII 2009, LNCS 5619, pp. 739–748, 2009.

properly for them. Recent approaches (e.g. del.icio.us) are apparently better accepted by the users but still have usability problems and are lacking some functionality [5].

The most important lesson learned from the failures of these tools is that the users' needs and preferences should not be neglected in the initial design of the tool. For this reason, we employed a user-centered design (UCD) [6] process which focuses on the users' needs as well as their tasks and goals.

2 User-Centered Design and Feature-Driven Development

To achieve a high usability, we started by interviewing researchers, educators, and students. The results were correlated to generate eleven prototypical use cases. Each use case describes the interaction between a user and the system in respect to achieving a certain goal.

2.1 Interviews

To get an impression about dealing with online resources, the following topics were discussed: current use of bookmarks, research for preparing lectures, lecturing and research workflows, social bookmarking with integrated contact management, teamwork, support for course studies, rating of potential features, and experience with existing solutions. Especially the first topic gives us hints to conclude the relevance and importance of the interviewees' statements.

The procedure and resulting answers of a verbal interview are not completely predictable [7]. An interviewee could, without being aware, talk a lot about a certain topic and this could take much more time as intended. Furthermore, unexpected but interesting topics could arise. Hence, we conducted semi-structured interviews (guided but not completely determined) to create an environment to facilitate fruitful interview sessions.

2.1.1 Searching, Storing and Recovering

One central question the interviewees were asked is how useful information can be found quickly and efficiently. How can we recover this information at a later point in time? How can we find similar information about this topic? The workflows of students, lecturers, and researchers were studied by focusing on their handling of web resources in order to conclude the best way to support them. This support allows them to quickly and efficiently give or attend lectures, create or complete assignments or lab work, do research, and communicate with colleagues. It was concluded that tagging bookmarks is much more efficient than filing them into a single category: If an individual wants to remember an item (article, image, book reference), they have to go through a multi-stage process [8]: After the individual makes a decision to remember something (stage 0), multiple concepts are usually activated simultaneously (stage 1), and then, one of these multiple concepts has to be chosen (stage 2): The information is then stored in the appropriate category. It might be difficult to access this information in the future, because the user has to remember what concept was finally chosen maybe months or even years ago. The underlying cognitive process for tagging differs from the process for categorizing, making the retrieval of information

relatively simple. The user can recover the bookmark by choosing any of the concepts that was activated in the tagging process.

2.1.2 Finding Something Similar

Frequently, the user likes to access related sources. It seems reasonable to provide an opportunity for users to benefit from the other users' knowledge. In a real world group of people, e. g. a business seminar, the participants of the seminar can share and communicate face-to-face. They know each other and their respective field of work.

Nevertheless, a conversation face-to-face or via e-mail is not always possible. Some interviewees tried to browse strangers' bookmarks, but they could not extract the information they needed. Because every user has a unique knowledge structure (cf. [9]) and every information structure created by this user depends on this unique knowledge structure, it is difficult to interpret other users' information structures without "knowing" this person. A better support of what Surowiecki [10] called the "wisdom of the crowd" in online communities seems to be obvious: "under the right circumstances, groups are remarkably intelligent and are often smarter than the smartest people in them". But how can this be exploited?

2.1.3 Information Gathering Workflows

Most people said that they often neither know the right space, nor the right time to save interesting information. It is also important to remember, that we often encounter information we have not actively searched for at that particular moment in time. Sometimes we come across interesting hints during work, which could be useful for some private interests. Then again, we find helpful articles for our job during surfing the web at home. Thus, the search results are still useful, but in a different context. Cutrell et al. [11] call this encountered information. This unexpected, but possibly useful information often interrupts the users' current workflows.

Moreover, it is possible to find multiple contexts by using certain tools. For example, somebody wants to call on a medical practitioner that a friend recommended a week ago. He does not remember the name or the telephone number of the practitioner, but he knows that he has saved it somewhere. Was it an email or an instant message? So, it must be in the inbox, or in his instant messaging logfile. Or, maybe it is already in his calendar's to-do-list. A useful PIM tool would deal with these multiple contexts e.g. by searching them automatically.

2.2 Workshop

Summing up, by studying all of the above-mentioned information, it is possible to determine meaningful use cases. For instance, "research new resources for lectures" or "find similar resources" can be considered as typical use cases. Based on these use cases, we derive 30 features for our tool, e. g. getting reminders of work to do or store bibliographical references in a repository. During a workshop with all interviewees, the features are ranked by importance. Afterwards, all features are grouped into feature sets. Each set consists of related features, i. e. all features of a set have to serve a joint purpose. These purposes are subject to dependencies, and therefore affect the order in which they can be put into practice.

2.3 Integrating Development Processes

By applying UCD, the potential users and their tasks are right in the center of the design process of the future system. Combining scientific information, users' opinions from interviews, information derived from the prioritization of the features and natural dependencies, a sorted list of feature sets emerge. To work through the list, we need a software development process which supports such an approach. The iterative and incremental process of feature-driven development (FDD) [12, 13] complements our UCD process in the development process.

Hence, we can start by implementing the most important features and quickly come to a running prototype, which then can be used and evaluated by the target users. The evaluation results are prioritized according to level of importance and when changes should be done. Less important features need not be dropped, but are left to be implemented later on in the development process.

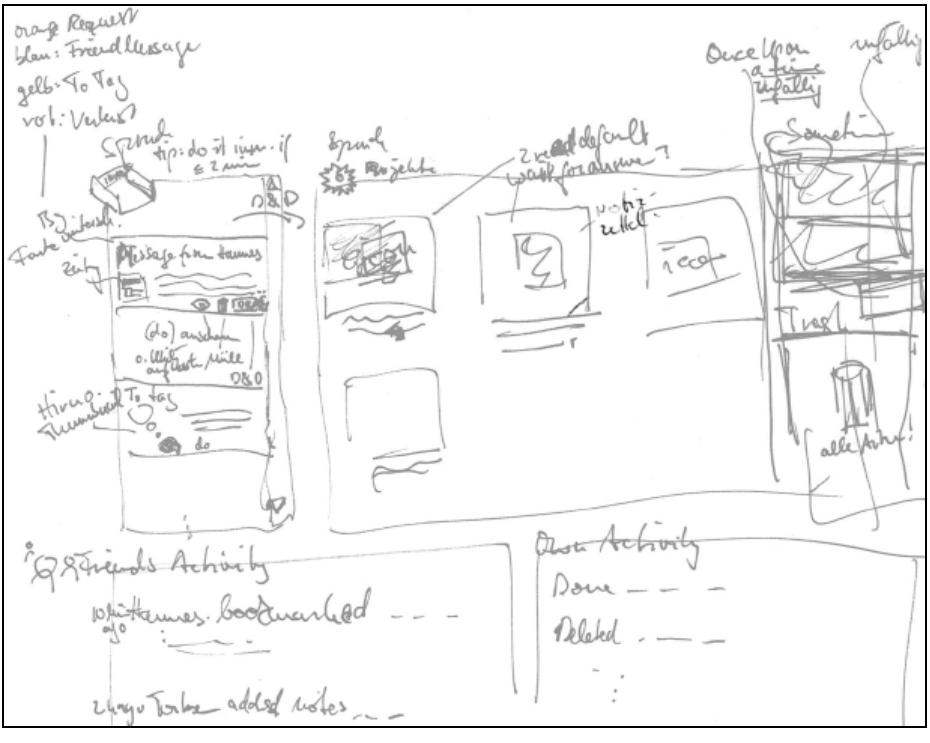

Fig. 1. Paper sketch of the dashboard

3 Design

Our myPIM tool consists of a core system and several modules. The core system implements the main functionality like entering, displaying, retrieving and distributing information. All further feature sets are implemented as modules, which can dock to

this core system. By using an FDD process as our software engineering paradigm, we were able to deploy the basic features of the core system quickly, without negative effect on the expandability of the system later on.

3.1 From Paper-Mockup to Final Version

According to the FDD process "plan by feature", we gear towards the feature sets and iterate through their features. Thus, for example, in the feature list "user interface and navigation" we start with a paper sketch of the interface (Fig.1) to get a general idea of the look and feel.

Adding more and more ideas leaves this sketch cluttered in the end. But, nevertheless, it is a useful instrument to organize the individual's thoughts. Screen mockups specify these ideas and lay the groundwork for the final implementation of the user interface (Fig. 2).

Fig. 2. Screen mockup of the signup dialog

3.2 Methodology's Benefit

By combining UCD and FDD, we can not only present first impressions of the look-and-feel, but also collect hands-on-experience of prospective community members at an early stage. This is especially important for systems that foster online communities because these communities only emerge under two conditions [14]. Either the system has to provide a so-called killer feature, which comes with immediate benefit for the users as soon as they use it – even if no one else uses it. Or, if no such feature can be implemented, the system has to provide awareness and so-called community memory [15, 16] for the users: (1) they need to be able to learn to get to know each other, (2)

they need to be able to see the others' histories, (3) they need to feel confident to meet each other again in the future within the community and (4) the code of conduct needs to be made obvious.

No matter if one aims to design for a killer feature or if one has to compensate for the lack of it, our methodology of combining UCD and FDD helps to iteratively create a product, which addresses the users' needs and their task requirements, and any defect can be counteracted in good time.

4 My Personal Information Manager

After presenting the integrated development process we introduce the core system basically with screen shots of the implemented main features.

Fig. 3. Dialog for signing up in three steps

4.1 User Interface and Navigation

A very important design goal was to allow the user to setup and work with myPIM instantly and easily. Therefore we designed the registering and setup procedure as a three-step process displayed on a single page (Fig. 3). Unlike plain HTML pages, many actions can be performed without (re-)loading a whole webpage. This asynchronous data transfer between browser and server is done with the help of the JavaScript frameworks Ajax, Prototype and script.aculo.us. For instance, the cross and the checkmark indicating the concept of failure and success are changed according to a continuous evaluation while the user is typing (cf. Fig. 3).

Moreover, there are three main areas: dashboard, bookmarks and community. On the dashboard, the user finds logs about the latest activities of his community's friends as well as of himself. All projects and the newest tasks that are to be completed can be seen on the dashboard. As summarized by Sinha [8], tags are good as fast accessible pointers to personal knowledge. On the other hand plain tags are not optimum for organizing, restructuring and working with knowledge. To counteract these limitations, we provided some basic project management functions for entities like bookmarks, notes and tasks.

The bookmark area is used to search both individual and other members' bookmarks, as well as to insert, edit and delete bookmarks. The user can customize the user avatar, title, tags, URL and bookmark listing system. To enhance community awareness, the latest comment regarding a certain bookmark is displayed in the large view.

The community area provides the standard functionality for maintaining relationships and tasks, e.g. friend requests, "breaking up friendships" as well as the list of conducted conversations.

4.2 Bookmarking and Workflow

To search for a bookmark, the user enters the search terms into the input field (Fig. 4), checks the desired search parameters (tags, title, notes, URL), specifies the scope (own, friends', and others' bookmarks) and hits the return key or clicks the button. The results are displayed in sorted order, which is specified in the upper right corner. For a fast search without visiting the myPIM web service first, users can add a searchlet to their browsers. This is usually done during the registration process. No matter which way myPIM is accessed, the search function always returns the same kind of results (Fig. 4).

4.3 Searching and Community

The principle of managing all relevant information in one context was combined with Allen's [17] time- and self-managing method called "Getting Things Done". With regard to the tasks that our system is designed to support, these activities deal with URLs, notes and messages from other users. myPIM provides context-related to-do lists so that users do not have to worry about forgetting something. The remaining

[1] A searchlet is a plugin that extends the browser based search functionality by adding a new search service to the browser search field.

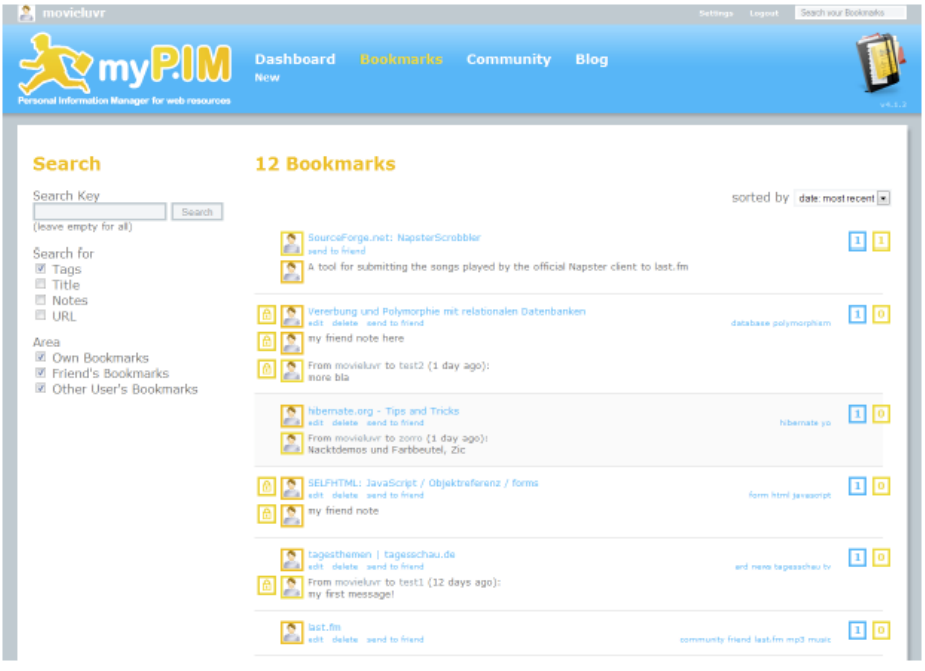

Fig. 4. Bookmark search and results

feature set "searching and community" complements our solution for the development of the tool myPIM. As stated above, we designed additional feature sets which are currently under development.

4.4 Future Work

So far we presented a description of the concept and the core features of our system myPIM. Now, we will briefly introduce further features that are part of the advanced system.

Because the supported resources of myPIM are manifold, they are made distinguishable by their types (e.g. PDF, web page, image) and can be sorted and grouped accordingly. The document types can be indicated by icons or thumbnail representations of the web resources. Furthermore, to save time, the resources to be displayed after a search takes place can be performed asynchronously on-the-fly.

The migration from previous (browser-based or online) bookmark archives to myPIM can be performed with an import function. All up-to-date browsers and online bookmarking services can export their bookmarks to a text file according to the Netscape standard. This file can be imported and integrated into the existing myPIM resource structure. If desired, the user can tag each bookmark within this process or create a reminder to do this at a later point in time.

To combine bookmarks and a file archive, we need a repository. During the bookmarking process, the bookmarked resource is saved in the repository and therefore

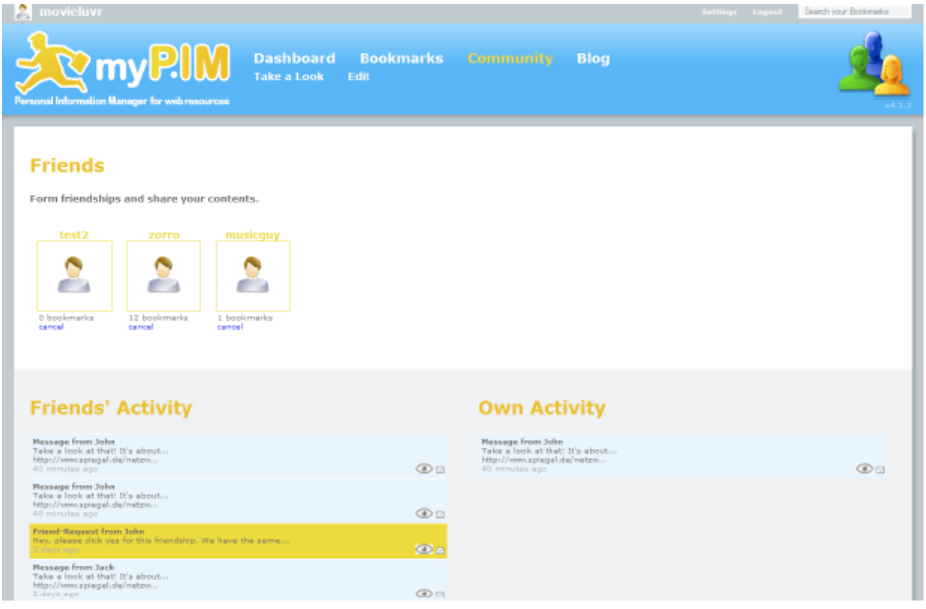

Fig. 5. Community dialog displaying friends and their activities (details negligible)

always available (mirroring). This makes local backups of web resources redundant. Additionally, the user could snapshoot the web page to freeze a certain state (versioning) [18].

5 Conclusion

The first evaluations show that not only the intended target audience of our system (computer scientists within research and teaching/learning context) can benefit from myPIM, but it can be easily extended to other target audiences. Our integrated UCD and FDD processes helped create a system than can be regarded as a general tool for users in all contexts.

References

1. Wittenburg, K., Das, D., Hill, W.C., Stead, L.: Group Asynchronous Browsing on the World Wide Web. In: 4th International World Wide Web Conference, pp. 51–62. O'Reilly, Boston (1995)
2. Marais, H., Bharat, K.: Supporting Cooperative and Personal Surfing with a Desktop Assistant. In: 10th Annual ACM Symposium on User Interface Software and Technology, pp. 129–138. ACM Press, New York (1997)
3. Keller, R.M., Wolfe, S.R., Chen, J.R., Rabinowitz, J.L., Mathe, N.: A Bookmarking Service for Organizing and Sharing URLs. Computer Networks and ISDN Systems, 1103–1114 (1997)

4. Kanawati, R., Malek, M.: A Multi-Agent System for Collaborative Bookmarking. In: 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1137–1138. ACM Press, Bologna (2002)
5. Hood, S.: del.icio.us Blog on Usability Lab, http://blog.del.icio.us/blog/2007/07/usability-lab.html
6. Holtzblatt, K.: Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design. Morgan Kaufmann, London (2005)
7. Bortz, J., Döring, N.: Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler. Springer, Heidelberg (2006)
8. Sinha, R.: A Cognitive Analysis of Tagging (or How the Lower Cognitive Cost of Tagging Makes It Popular). Thoughts on Technology, Design & Cognition, http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html
9. Krohn, T., Kindsmüller, M.C., Herczeg, M.: myPIM: A Graphical Information Management System for Web Resources. In: Ågerfalk, P.J., Delugach, H., Lind, M. (eds.) 3rd International Conference on Pragmatic Web. ICPW 2008, vol. 363, pp. 3–12. ACM, New York (2008)
10. Surowiecki, J.: The Wisdom of Crowds. Random House, New York (2005)
11. Cutrell, E., Dumais, S., Teevan, J.: Searching to Eliminate Personal Information Management. Communications of the ACM Archive 49(1), 58–64 (2006)
12. Coad, P., Lefebvre, E., De Luca, J.: Java Modeling in Color With UML: Enterprise Components and Process. Prentice Hall, Upper Saddle River (1999)
13. Palmer, S.R., Felsing, J.M.: A Practical Guide to Feature-Driven Development. Prentice-Hall, Upper Saddle River (2002)
14. Kindsmüller, M.C., Melzer, A., Mentler, T.: Online Community Building. In: Khosrow-Pour, M. (ed.) Encyclopedia of Information Science and Technology, pp. 2899–2905. Information Science Publishing, Hershey (2009)
15. Kollock, P.: The Economies of Online Cooperation. Gifts and Public Goods in Cyberspace. In: Smith, M.A., Kollock, P. (eds.) Communities in Cyberspace. Routledge, London (1999)
16. Kollock, P., Smith, M.: Managing the Virtual Commons: Cooperation and Conflict in Computer Communities. In: Herring, S. (ed.) Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives, pp. 109–128. John Benjamins, Amsterdam (1996)
17. Allen, D.: Getting Things Done. The Art of Stress-Free Productivity. Penguin, London (2003)
18. Reinke, C.: Optimierung von Reload-Strategien im Web durch Komponenten-Tracking. Institute for Informationsystems, University of Lübeck (2007)