INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

A CLASSIFICATION SCHEMA FOR PROCESS AND METHOD ADAPTATION IN SOFTWARE DESIGN PROJECTS

A. Ploskonos and M. Uflacker

Keywords: design planning, process improvement, user-centered design, process adaptation

1. Introduction

The integration of user-centered design (UCD) principles into software development processes has achieved much momentum across the software industry. The benefits of a design-driven, end-user focused approach in the development of complex interactive software systems are self-evident: an increase in usability, user satisfaction and total end-user experience are generally expected results. Many design process theories and methodologies have been developed to address this issue and have reached certain prominence in the HCI community and among software practitioners (e.g. [Holtzblatt 1998, Vredenburg 2001, Sharp 2007, Cooper 2003]). However, realigning general UCD approaches to real software design projects remains an inevitable, yet challenging task. A reasonable selection of design methods and adjustments before and during the design iterations is required. By focusing on the individual scope and goals of a particular project, better decisions about what kind of methods to select can be made [Norman 2002]. Still, contextual process requirements remain mostly unconsidered in the literature. Due to the generality often found in design process theories and methods, the adaptation and assessment of processes and design practices in the context of their deployment is often left to the experience of stakeholders. Vredenburg points out that "*a rigorous end-to-end methodology is not being practiced yet*", and that in order to be effective, "*such a methodology should be scalable based on project characteristics*" [Vredenburg 2005].

In this context, we have observed design-intensive projects and analyzed differences and commonalities in regard to project characteristics, the applied process, activities and chosen design methods. Building on these observations, a classification schema for design projects is presented in this paper. The schema suggests four types of design projects along two dimensions: initial awareness of end-user needs and scope of the aspired solution. The four project categories are discussed in terms of process characteristics, design activities and recommendations for UCD method selection.

The catalyst for our research was the work of Scanlon and Percival [Scanlon 2002], in which the applicability of UCD core activities across a range of development project types is described. The authors distinguish four types of projects: customization of vendor applications, evolution of existing applications, rewriting existing applications, and the development of new applications. Their definition of project types was based on frequently encountered projects in software product lifecycles rather than the nature of applied design processes, which we think is more appropriate for the adaptation of a design methodology. The appraisal of best practices for process and method adaptation according to representative key features of a project's context can facilitate the planning and management of design projects.

2. Field Study

During our research we wanted to discover and identify different types of design projects in order to narrow down adequate process implementations and design techniques for each type. For this purpose, we have observed and analyzed eight different projects settled in the San Francisco bay area. We aimed for a broad distribution of projects as measured by project parameters such as thematic classification, duration and team size. The projects were carried out by stakeholders from educational and research institutions up to the commercial domain. Thus, we achieved an adequate level of variety and diversity to avoid an unwanted focus on a particular design sector. Selected project tasks were placed in the business software domain, consumer and e-government applications, as well as product design. Specifically, we observed design practitioners working on distinct tasks such as the development of a software solution for sales pipeline management and brand management, the development of a new solution for improving community development in the context of e-government, the development of a software portal for executives combing strategy, planning, customer, and market perspectives, or the usability improvement of an application used for handling customer requests. Common requirement for all selected projects was a strong focus on founded design methodologies and a user-centered approach. The team sizes ranged between 3 and 15 persons directly involved in the design process, whereas the average size of the design teams was 9 persons. The duration of a project averaged to 5 months, with a minimum of 3 weeks and a maximum of 12 months.

The projects have been either observed in detail during the course or analyzed after completion by interviewing team members and studying project documentations. On top of this, project characteristics, design processes, and method overviews were elaborated. For the interviews, we asked process participants to give a detailed description of the project course, tasks, and the produced solution. We were also focusing on the experienced problems during the project, the team structure, timeline of the project, and furthermore asked to draw and describe the design process including its milestones and used design instruments. Another source of information were documents that have been generated during or at the end of a project. We looked at initial descriptions of the project including tasks, plans and team members, prototype descriptions and demos, and reports of research studies being done during the project, such as market and competition studies, or technical research. Again, we conducted interviews in order to confirm our understanding and clarify the points. The next step was to analyze the projects and their context, for which we explored the phases of the design process, used methods and project characteristics including aimed tasks and solution. Then we compared the projects in terms of similarities and differences. From the comparison of project characteristics we received insights for a classification schema and from the models of design process and methods we extracted common process and method characteristics for each project type.

3. Design project classification

From our observation we inferred two satisfying dimensions that we could use to describe the nature of a design project. The results showed that design process organization and the selection of design techniques primarily depend on the initial awareness level of user needs and the scope of the sought-after solutions. At the beginning of a project, user needs can be classified as either being *explicit* or *implicit*. User needs are explicit if they can be readily articulated by the users. That is, users have a good understanding of what they need to meet their requirements, e.g. due to previous experience with similar products. Implicit user needs in contrast have to be discovered by the design team. Initially in this case, users are not aware of any usability issues or possibilities. We further noticed that this dimension strongly correlates to the task given to a project: a well-defined task at the beginning of a project often infers a high level of user awareness, whereas an ambiguous task requires more efforts in discovering what the users really need. In the solution dimension we differentiate between the development of a *new* solution and the *improvement* of an existing solution. As a result, our schema defines four different types of design projects: *Usability*, *Capability*, *Extension*, and *Innovation* projects, as shown in Figure 1.

3.1 Usability projects

Usability projects are characterized by explicit end-user needs from the outset. These projects deal with already existing software solutions in use, but which are apparently affected by problems concerning their usefulness and usability. The project goal is to re-design the existing solution in order to improve the usability of the application. Consequently, the main task for *Usability* projects is to discover problems in the implementation of the system, to optimize the user experience, and to discover better-aligned patterns for the user interface to optimally support the user needs. Users know the software application very well and can say exactly where and why they have problems using it. The task of a designer is to improve the overall usability of the system and to enhance the user experience, e.g. by improving the navigation and the flow of screens, or optimizing the number of required user interactions with the system. Another task typical for *Usability* projects is the improvement of the systems' backend. This can also impact on user experience, e.g. in terms of achieving better performance, response times, etc. If necessary, the project might also include the task to add new functions that have been left out in previous versions of the application. One example for a *Usability* project that we have analysed was the usability improvement of a software application for handling customer requests.

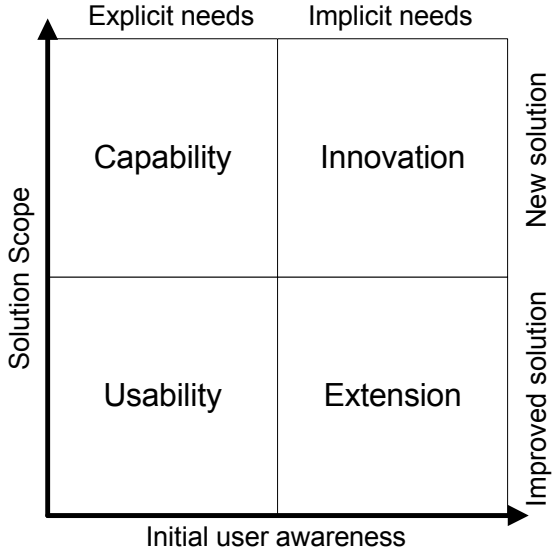

Figure 1. A Classification Schema for Design Projects

3.2 Capability projects

When a project team is confronted with explicit user needs and is assigned to design a new suitable system, it is categorized as a *Capability* project. A typical task is to develop a new software solution basically from scratch, but for which user requirements are already well understood. This is because already existing solutions have been previously used by the targeted user groups to fulfill certain needs, but the features were not satisfying or suitable for the user tasks at hand. In other words, previous attempts to solve the user needs have been made, but they did not achieve appropriate results. Users are still struggling with some problems, for example, the existing application cannot treat a big amount of information, or the application has performance issues, etc. Another reason could be that existing applications are too expensive and the task of the project is to develop a coequal application with lower costs. Thus, the result of a *Capability* project is a new solution that is more capable, i.e. performs better than other solutions in various respects and for a given set of user needs. Among the

observed projects that fell into this category was the development of a new portal for executives combining strategy, planning, and customer and marketing perspectives for running businesses more effectively, and the development of a new software application for sales pipeline and brand management.

3.3 Extension projects

Extension projects differ from the previous two project categories in regard to the initial awareness of the end-user needs. User requirements are concealed, while the project aims to improve the user experience of an existing solution. The question how this can be achieved is completely unanswered at the beginning of a project. Furthermore, *Extension* projects are often driven by new technologies. As new technologies emerge, new opportunities arise for extending existing applications with innovative features. The users have not yet realized that they can benefit from a system extension that now has become feasible. Those extensions can bring additional benefits to the user experience. The task of a designer is to discover implicit user needs and to find out how additional features can improve the user's work. Examples for this type of project was the enhancement of a customer relation management application with Web 2.0 features and the improvement of a product customization service .

3.4 Innovation projects

Projects dealing with implicit user needs and the design of a new solution fall into the fourth category of *Innovation* projects. The task for those projects is mostly undefined or vague in order to leave space for unrestricted findings. Designers start on a greenfield without exactly knowing what they are looking for. They apply processes and methods to unfold new opportunities. The goal is to discover implicit needs of users, develop an innovative solution that satisfies those needs, and to present a new system on a possibly new market. *Innovation* projects intend to bring innovative and useful ideas to the users' lives or work practices. Thus, these types of projects tend to be challenging and generally hard to predict in terms of the design process and method selection. One example for this type of projects was the development of a new application supporting community development in the context of e-government.

4. Process characteristics

To guide in the development of viable, feasible, and enjoyable software, a user-centered design process generally embraces the following activities. Starting with the basic requirement to *understand* the problem, the team discovers related knowledge and looks at the world in order to define what has already been done. This may involve close examination of the targeted problem domain and state-of-the-art technologies. The next phase is to *observe* users in order to discover their needs and experience their lives and working behaviour. After gathering a variety of information teams start to *synthesize* user requirements, user needs and to create an explicit statement of the problem from a defined point of view. The next step is to *ideate* as many solutions as possible. Ideas can be unrealistic and broad, since realization is out of scope in this phase. The goal is to diverge again in many different directions. In the next phase *prototypes* are developed for the experimental implementation of promising ideas. Such a prototype shows whether an idea works out as expected, exposes unanticipated issues and reveals new opportunities that have been overseen at the early stage. Prototypes are later *validated* together with real end-users in order to obtain necessary feedback for solution improvement.
This rough sketch of a design process outlines the fundamental parts of a user-centered design methodology. Actual design projects, combine these core activities in a non-linear fashion and act on a highly iterative process with an undetermined number of loops between often overlapping and blurred phases. The exact specification of an individual design process is therefore diverse and depends on several project parameters, as discussed. Studying the characteristics of different design projects, we discovered common process structures and activities for each type of design project and present those in the following.

4.1 Process characteristics for *Usability* projects

Since *Usability* projects are dealing with existing applications, the activities start with an analysis of those systems in terms of functionality, usability, correctness, efficiency, intuitiveness and other critical factors. Information that has been collected in previous user research sessions may be already available from preceding projects but should be checked and confirmed in the new context. The next steps in this type of projects involve the observation of users working with the system and to record usability problems and user perceptions. The explicitness of user needs in this kind of projects renders these activities relatively straightforward and linear. The collected results of the user research in the *Understand* and *Observe* phases are analyzed by the team and interpreted in order to define explicit user characteristics and needs. Teams systemize information about user experience of working with the system and prioritize user requirements. Once the users' characteristics and needs are defined, the project starts to re-design the system with the goal to improve its overall usability. One of the first activities within the *Ideate* phase is to develop a new vision of working with the system and to evaluate it with end-users. After having elaborated a validated concept, user interfaces, flow of screens, and visual designs can be developed. Decisions on navigational concepts, information flow, (what is clickable, what is next, etc.), are made. Ideas developed in this design phase are prototyped using appropriate tools and evaluated by users during the *Validate* phase.

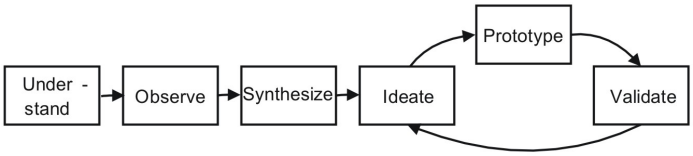

Figure 2. Process iterations in *Usability* projects

The typical design process that we extracted from observing *Usability* projects is shown in Figure 2. The main characteristic is that the *Understand*, *Observe*, and *Synthesize* phase follow one after another in a relatively linear way. The design team conducts user research with minimal or no iterations because users can readily express what their needs are and where they are facing problems with the existing system. A thorough discovery of the problem domain is therefore completed with relative ease. Furthermore, these requirements do rarely change during *Usability* projects. The user involvement is intensified during the design of the system, and the consecutive prototyping and validation phases. Here, iterations continue until a solution is found that meets the identified needs.

4.2 Process Characteristics for *Capability* Projects

Capability projects set out to develop a new application for established processes that are performed manually or by using some existing tools in place. Early project activities include market studies and investigations of related and competitive products, where the focus lies on implemented functionality, user interfaces, advantages and disadvantages of the system. Next steps involve the observation of work processes, performed tasks, user environments and potential collaboration habits. After a clear definition of user needs, the sketching of design approaches helps to define a concept of working with the envisioned system. Major activities during *Ideation* comprise also the design of core elements of the user interface and the software architecture of the new system. Finally, ideas are prototyped and validated with the end-users and other relevant stakeholders. Multiple iterations allow for a continuous evolution of the design. Starting with rapid paper prototyping of user interfaces, teams continue to prototype more detailed visual designs and finish with a functional prototype that gives users the chance to experience the work with the system. The design process discovered during our observation of *Capability* projects is presented in Figure 3. The process iterations are close to those found in *Usability* projects because *Capability* projects also benefit from explicit user needs. One difference is that designing a new solution and prototyping ideas requires more team effort and usually more iterations, since the task of *Capability* projects is to develop a new product that did not exist before on

the market. In many cases, the design of a new product requires intensive prototyping and quick iterated user validation before discovering an appropriate solution.

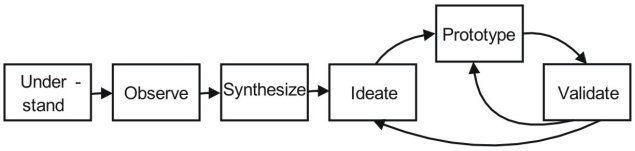

Figure 3. Process iterations in *Capability* projects

4.3 Process Characteristics for *Extension* Projects

Since *Extension* projects seek after innovative extensions for existing solutions (often inspired by new technologies), the first project activities in this category aim to understand and evaluate the existing systems under examination, new technologies and the business context. Other important aspects target the observation of users in order to discover implicit needs, tasks and unconsidered processes. To inspire the design team, future user scenarios are developed, synthesizing all findings from previous domain research. Based on the new scenarios, *Extension* projects can now continue with the iterated design and prototyping of the system to imagine how it might behave and refine it through continuous user validation with the goal to discover and support extended working patterns. The purpose of prototyping for this type of project is first to test initial concepts in order to discover unanticipated issues and new opportunities, then to explore new functionality, and finally, to prototype a final design implementing new features of the extension. If the solution involves the use of new technologies, the development of a technical prototype for software tests helps to ensure the feasibility of the system.

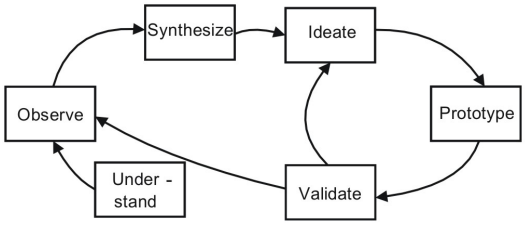

Figure 4. Process iterations in *Extension* projects

A typical design process for *Extension* projects is shown in Figure 4. To discover implicit user needs, an iterative user research is indispensable. A detailed identification of user needs can generally not be achieved after one user research session and requires more efforts and repeated observations. After a validation of ideas with users, the team can come up with new insights or defects in the design that require to iterate and continue the user research. Having sufficiently identified the user needs, the next iterative cycle is to continuously design and re-design a solution, to prototype, and to validate with users until having reached a state that satisfies the requirements.

4.4 Process Characteristics for *Innovation* Projects

Innovation projects begin with extensive benchmarking activities – the study of the situation in the industry, used technologies, and related products in order to define a first direction of the project. However, it can be expected that these initial directions are often subject to change during an *Innovation* project. The development of an innovative solution primarily depends on a thorough definition of user needs and context. That is why iterative need-finding is emphasized in those projects until implicit user needs are sufficiently discovered. Insights are sketched and validated right from the start, ideas are rapidly prototyped in order to test with users and obtain valuable feedback for

refinement. Once project directions, user characteristics, and future usage scenarios have been defined, next phases involve prototyping system functionality and the system architecture. To prove all design ideas, the teams build prototypes with increasing fidelity, continuously approaching the final solution. The design process of *Innovation* projects is similar to those of *Extension* projects in terms of discovering implicit user needs. However, the process for *Innovation* projects is much more complex, iterative, and unpredictable overall. After validation of ideas with users, the process can turn again to the *Understand* phase in case of design defects, or turn to the *Observe* phase in case of required additional user research, or turn to the *Synthesize* phase in case of changing user requirements (see Figure 5).

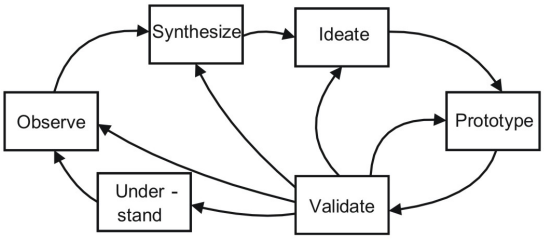

Figure 5. Process iterations in *Innovation* projects

5. Design methods in context

During our observation of different design projects we paid attention to common methods the teams applied during the course of designing. We systemized the gathered information and found patterns indicating that certain types of design methods are commonly used in every project while other methods are appropriate only for specific project types. Subject matter expert interviews [Cooper 2003] were conducted at the beginning of each project. The teams looked for experts in the investigated subject and contacted them in order to get the first introduction into the subject and define directions for further research. Competitive research and literature reviews helped to define advantages and disadvantages of competing approaches. The teams took into account this information and incorporated it in their design of a new solution. Other very popular methods were contextual user observations and interviews [Cooper 2003, Holtzblatt 1998]. By observing or interviewing users in context, observers paid attention to user activities, the environment, user interactions, used objects, and user characteristics. Recording tools such as note taking, audio recording, taking photographs, and video recording were applied. The teams used those tools to explore undiscovered patterns of observed subjects. To further inspire the design team and to stimulate design thinking, another common activity was the saturation of design space, where the teams put all artefacts and findings on a wall (pictures, quotes, notes, etc.). Members of the team told their user story during interpretation sessions, while others capture and share their notes. Within an interpretation session the teams often used affinity diagrams or mindmaps to systemize all findings and to find design principles [Holtzblatt 1998]. Some teams used personas or simple lists of user characteristics for defining a user. Each project also conducted brainstorming sessions [e.g. Wilson 2006] to produce a variety of diverging ideas. To better depict how the new system will work, storyboards were drawn on flipcharts. Based on the sketched storyboard teams collaboratively developed detailed scenarios [e.g. Sharp 2007] of using a new system in order to share their perspective with others. Paper prototypes provided a quick and convenient way to sketch various screens and to quickly get user feedback.

We think that these methods are commonly used because they are easily applied and refer to basic UCD principles of iterative, human-centered, and prototype-driven approaches. The next step of our research was to find differences between selected methods for project types and to reveal common characteristics for methods within the project group. For this purpose we created affinity diagrams by clustering the observed methods by project types in order to find similarities and differences. The discovered patterns are illustrated in Figure 6 and presented in the following.

5.1 Design methods in *Usability* projects

Design methods for *Usability* projects are oriented on helping designers understand and analyze current user work – how they work, kinds of performed tasks and so on (for example focus group, hierarchical task analyses [Sharp 2007], work models [Holtzblatt 1998]). Another aspect is that design methods are also oriented to help to analyze software applications in terms of discovering functional problems, usability problems, and other problems in software application (e.g. eye tracking).

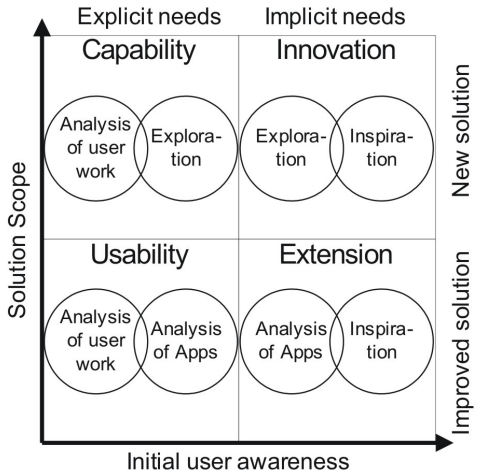

Figure 6. Clusters of design methods for different project types

By carrying out a usability tests [e.g. Dumas 1999] an observer paid attention to implemented functionality, correctness, learnability of the system, efficiency, usability, visual design, intuitiveness, flow of screens, terminology, and pain points. In order to obtain valuable feedback from usability experts, the application was analyzed by applying heuristic evaluation [Nielsen 2005]. Where the investigated application had a large number of users, surveys and questionnaires [Sharp 2007] have proven to be useful in obtaining usability perceptions and opinions from a wide range of users. To further understand how people process information and how to present information more effectively, *Usability* projects used cognitive human factors [Sharp 2007].

5.2 Design methods in *Capability* projects

The methods selected specifically for *Capability* projects guided in the analysis of current user work and help to explore new opportunities for a new system (using e.g. metaphors, mindmaps, rapid prototyping).

Since Capability projects assume development of a new application, the first method commonly used was a historical analysis of industry, or organization, or market in order to define trends and patterns of development and to make a forecast of the future. The team used this method to define the right strategy for the new product. Another task for this kind of project was to understand the current user work and all related issues. Thus, the teams studied business processes, the culture in the organization and social networks. To systemize findings about user work, a work model [Holtzblatt 1998] or detailed use cases were build. Hierarchical task analyses [Sharp 2007] help to make deeper investigations into user tasks and to reveal dependences and common structures.

Another method often used by the design teams was to ask the user to draw the experience of working with the system in order to obtain user attitudes towards design problems and to gather opinions concerning a new system. Like in *Usability* projects, cognitive human factors were utilized in order to define how to present information in an appropriate way. To test design ideas, testing script [Sharp 2007], usability tests and heuristic reviews were used in this kind of project.

5.3 Design methods in *Extension* projects

For *Extension* projects teams specifically applied methods to support the analysis of the application under investigation, in order to reveal hidden issues and new opportunities for improvement. This helped to inspire the collective team and to explore innovative extensions for the application (such as role playing [Buxton 2007], experience prototyping [Buchenau 2000], storytelling, and so on).
Power of ten is a technique to take a different perspective on the design problem: if you have been looking at the individual experience, now change your frame and look at a group experience. The team changed observational perspectives in order to obtain additional useful insights. Metaphors were defined for the design problem and used and look at the metaphor instead to observe and reveal how things work. In extreme user interviews researchers observed and interviewed those users, who use a product in terms of extreme quantity. Their experiences were valuable for the design and the team obtained new insights for using the system. Within the projects, teams used sketching to develop many concepts for extension (e.g. [Buxton 2007]. Sketches were used to explore new ideas, to share them, and to suggest improvements. Another method for *Extension* projects was role-playing, which has been conducted to experience the use of a system and to inspire for new ideas. This valuable experience could not be achieved by mere user observations or interviews.

5.4 Design methods in *Innovation* projects

For *Innovative* projects selected methods helped to explore new opportunities for design – see insights beyond the obvious – and to inspire the team for innovative design. The development of a new product requires exploration tools and inspiration tools, supporting in the challenging task to build a completely new system, where it is impossible to compare or to touch.
Since *Innovation* projects deal with the design of a new solution just like *Capability* projects, a commonly used method here was to conduct a historical analysis of the industry or the society to predict social and technological developments, as well as future needs. Naturally, project exploration tools such as power of ten, metaphor, and extreme user interviews have proven to be extremely helpful for this kind of projects, as they all help to discover new opportunities for design. In addition a competitive analog was commonly utilized. By observing one subject additional insights could be discovered for a competitive subject.
Within *Innovation* projects to reveal new opportunities for design or user behaviours in context, designers spent much time by only watching daily user work without interactions with the user ("fly on the wall"). The observer gives herself the time to watch for a while, sees things she would expect, watches some more, until she suddenly discovers something she did not expect. Sketching and role-playing were also used for *Innovation* projects in order to explore unanticipated issues and to experience the life of users. To share design ideas within the team stakeholders, storytelling was another popular method. It helped to express important aspects of a new design and to sell ideas because people are generally in favour of listening to and telling a story.

6. Conclusion

We have presented a classification schema for software design projects in order to support process and method adaptation. The need for this classification has stemmed from the difficulty to align generic process descriptions and methods in UCD with the contextual requirements of individual design projects. We started with an observation of eight different design projects and analyzed the applied design processes, core activities, and methods. Based on these insights, we defined a schema to classify design projects into the four project categories *Usability*, *Capability*, *Extension*, and *Innovation*. The two dimensions along which the projects have been classified were (1.) the initial awareness of end-user needs (being either implicit or explicit), and (2.) the scope of the solution (being a new system or the improvement of an existing one). These metrics turned out to be well chosen, as they can be generically applied to design projects in their earliest phase. The schema allows stakeholders to classify projects and to early obtain information on how to set up the design process and plan for activities. This information has been given in form of process characteristics that we have observed for each type of project. We further elaborated our observational study by presenting

recommendations for method selection that have turned out to notably support the design process in a particular project context. Our field study has provided first data and insights that were structured and presented in this schema context. More data would further enhance the schema applicability, thus motivating to extend our field studies with further experiences. The two dimensions that were proposed for this classification provide a solid, yet simple schema. Future extensions might include time and budget dimensions that have been left out here. We believe that this work contributes to the integration of UCD perspectives into product development, as it helps designers and project stakeholders to tailor the design process more accurately and to adapt their activities to the design context at hand.

References

Buchenau, M. and Suri, J,. "Experience prototyping", Proceedings of the Conference on Designing Interactive Systems: Process, Practices, Methods, and Tchniques,ACM New York, 2000, 424-433.

Buxton, B., "Sketching User Experience: Getting the Design Right and the Right Design", Morgan Kaufmann Publishers San Francisco CA,2007.

Cooper, A. and Reimann, R., "About Face 2.0: The Essentials of Interaction Design", Wiley Publishing, Inc, Indianapolis, 2003.

Dumas, J. S. and Redish, J. C., "A Practical Guide to Usability Testing", Intellect Books England, 1999.

Holtzblatt, K. and Beyer, H., "Contextual design: Defining Customer-Centered Systems", Morgan Kaufmann Publishers San Francisco, 1998.

Nielsen, J. "Heuristic Evaluation", 2005, http://www.useit.com/papers/heuristic/. Accessed Sep. 14th, 2007.

Norman, D. A., "The Design of Everyday Things", Basic Book New York, 2002.

Scanlon, J. and Percival, L., "UCD for different project types", IBM developerworks, 2002, http://www.ibm.com/developerworks/library/us-ucd/, Accessed Nov. 14th, 2007.

Sharp, H., Rogers, Y. and Preece, J., "Interaction Design: Beyond Human-Computer Interaction", John Wiley and Sons New York, 2007.

Vredenburg, K., Isensee, S, and Righi, C., "User-Centered Design: An Integrated Approach", Prentice Hall, 2001.

Vredenburg, K. Mao, J. Smith, P. W. and Carey, T., "The state of user-centered design practice", ACM, volume 48, 2005, 106—109.

Wilson, C., "The well-tempered practitioner: Brainstorming pitfalls and best practices" ACM Press, Volume 13, 2006, 50 – 63.

Anna Ploskonos, B.Sc.
Hasso-Plattner-Institute for IT Systems Engineering, School of Design Thinking
August-Bebel-Str. 88, 14482 Potsdam, Germany
Tel.: +49 331 97992-566
Fax.: +49 331 97992-579
Email: anna.ploskonos@student.hpi.uni-potsdam.de