

# An Interdisciplinary Approach for Successfully Integrating Human-Centered Design Methods into Development Processes Practiced by Industrial Software Development Organizations

Eduard Metzker and Michael Offergeld

DaimlerChrysler Research and Technology Center Ulm, Software Technology Lab,  
P.O. Box 2360,  
D-89013 Ulm, Germany  
{eduard.metzker, michael.offergeld}@daimlerchrysler.com

**Abstract.** In a world where competitors are just a mouse-click away, human-centered design (HCD) methods change from a last minute add-on to a vital part of the software development lifecycle. However, case studies indicate that existing process models for HCD are not prepared to cope with the organizational obstacles typically encountered during the introduction and establishment of HCD methods in industrial software development organizations. Knowledge about exactly how to most efficiently and smoothly integrate HCD methods into development processes practiced by software development organizations is still not available. To bridge this gap, we present the experience-based human-centered design lifecycle, an interdisciplinary effort of experts in the fields of software engineering, human-computer interaction, and process improvement. Our approach aims at supporting the introduction, establishment and continuous improvement of HCD processes in software development organizations. The approach comprises a process model, tools, and organizational measures that promote the utilization of HCD methods in otherwise technology-centered development processes and facilitate organizational learning in HCD. We present results of a case study where our approach has been successfully applied in a major industrial software development project.

## 1 Introduction

The relevance of usability as a software quality factor is continually increasing within the industrial software development community. One fact contributing to this trend is the bringing into being of new regulations and standards for software usability that are increasingly influencing national legislation, such as the *EC Directive about Safety and Health of VDU Work* [1, 2]. Contractors tend to demand compliance with these regulations, thus forcing development organizations to adopt human-centered design (HCD) approaches in order to achieve defined usability goals. Another point is the still growing market for commercial web-based services such as on-line banking applications or e-commerce portals: The success of these systems depends even more on usability and user satisfaction than that of traditional desktop applications since

their easy availability via WWW (world wide web) enforces competition between their provider organizations [3]. The very same effects will carry over to the mobile, proactive services embedded in the information appliances of the near future [4].

It is well accepted both among software practitioners and in the human-computer interaction research community that structured approaches and explicit HCD engineering processes are required to build systems with high usability [5-11]. However, a closer look at documented case studies of existing HCD process models shows that these approaches do not provide solutions for the severe organizational obstacles that are encountered in establishing HCD methods in the development processes practiced by industrial software development organizations (see below).

To overcome these shortcomings we developed the experience-based HCD lifecycle, a result of a concerted effort of practitioners and researchers in the fields of software engineering, human computer interaction and process improvement. Our approach is designed to improve the utilization of HCD methods in standard software development processes and facilitate organizational learning concerning HCD activities. We explain the experience-based HCD lifecycle by describing the respective process models, tools and organizational measures that constitute our approach. We describe how our approach was verified during a long-time evaluation where it was successfully applied to improve the development process of an interactive software system. Finally we present the results of this evaluation and the lessons learned.

## 2 Existing Process Models for Human-Centered Design

There is a large body of research and practical experience available on software process models which are used to describe and manage the development process of software systems. Prominent examples are the *waterfall model* [12], the *spiral model* [13], or the *fountain model* [14]. Yet for the shortcomings of traditional process models concerning usability issues, a number of approaches have been developed that take into account the special problems encountered with the development of highly interactive systems [6, 10, 11, 15, 16]. According to ISO13407 these approaches can be embraced by the term ‘human-centered design processes’ [17]. In this section we focus on those approaches which have been extensively applied to industrial software development projects. We outline their basic principles and discuss some of their drawbacks based on documented case studies.

One of the first approaches used to address usability issues was the soft system methodology (SSM) [15, 18]. SSM was widely applied to capture the objectives, people involved (e.g. stakeholder, actors, and clients), constraints, and different views of interactive systems during development. However, since SSM’s origins are in general systems theory, rather than computer science, it lacks many of the specific HCD activities such as construction of user interface mockups or iterative usability testing which are necessary to fully specify interactive systems. These shortcomings limit the utilization of SSM to the early activities of the development process such as requirements or task analysis.

The star lifecycle [16], proposed by Hix and Hartson, focuses on usability evaluation as the central process activity. Around this central task the, activities

system / task / functional / user analysis, requirements / usability specifications, design & design representation, rapid prototyping, software production and deployment are placed. The results of each activity such as task analysis are subjected to an evaluation before going on to the next process activity. The bi-directional links between the central usability evaluation task and all other process activities cause the graphical representation of the model to look like a star.

One problem concerning this approach was already outlined by Hix and Hartson [16]: Project managers tend to have problems with the highly iterative nature of the model. They find it difficult to decide when a specific iteration is completed, complicating the management of resources and limiting their ability to control the overall progress of the development process. Furthermore, the star lifecycle addresses only the interactive parts of a software system, leaving open how to integrate the star lifecycle with a general software development method.

The usability engineering lifecycle [10] is an attempt to redesign the whole software development process around usability engineering knowledge, methods, and activities. This process starts with a structured requirements analysis concerning usability issues. The data gathered from the requirements analysis is used to define explicit, measurable usability goals of the proposed system. The usability engineering lifecycle focuses on accomplishing the defined usability goals using an iteration of usability engineering methods such as conceptual model design, user interface mockups, prototyping and usability testing [6]. The iterative process is finished if the usability goals have been met.

As outlined by Mayhew [10], the usability engineering lifecycle has been successfully applied throughout various projects. However, some general drawbacks have been discovered by Mayhew during these case studies: One important concern is that redesigning the whole development process around usability issues often poses a problem regarding the organizational culture of software development organizations. The well established development processes of an organization can not be turned into human-centered processes during a single project. Furthermore, the knowledge necessary to perform the HCD activities is often missing in the development teams, hampering the persistent establishment of HCD activities within the practiced development processes. How the HCD activities proposed in the usability engineering lifecycle should be integrated exactly and smoothly into development processes practiced by software development organizations, was declared by Mayhew as an open research issue [10].

Usage-centered design [11], developed by Constantine and Lockwood, is based on a process model called activity model for usage-centered design. The activity model describes a concurrent HCD process starting with the activities of collaborative requirements modeling, task modeling, and domain modeling, in order to elicit basic requirements of the planned software system. The requirements analysis phase is followed by the design activities: interface content modeling and implementation modeling. These activities are continuously repeated until the system passes the usability inspections carried out after each iteration. The design and test activities are paralleled by help system / documentation development and standards / style definition for the proposed system. This general framework of activities is supplemented by special methods like essential use case models or user role maps.

Constantine and Lockwood provide many case studies where usage centered design was successfully applied, yet they basically encountered the same organizational obstacles as Mayhew [10] when introducing their HCD approach into

software development processes practiced. They emphasize the fact that ‘new practices, processes, and tools have to be introduced into the organization and then spread beyond the point of introduction’ [11]. A straightforward solution to these problems is training courses for all participants of HCD activities offered by external consultants. However, this solution is regarded as being time consuming and cost intensive in the long run. It tends to have only a limited temporary effect and thus does not promote organizational learning on HCD design methods [11]. Constantine and Lockwood conclude that it is necessary to build up an internal body of knowledge concerning HCD methods, best practices and tools tailored to the needs of the development organization.

As outlined above, the approaches described have some limitations in common which hamper their effective application and establishment in an industrial setting. These shortcomings can be summarized as follows:

- Existing HCD processes are decoupled from the overall system development process.

One common concern relating to these approaches is that they are regarded by software project managers as being somehow decoupled from the software development process practiced by the development teams. It appears to project managers that they have to control two separate processes: the overall system development process and the HCD process for the interactive components. As it remains unclear how to integrate and manage both perspectives, the HCD activities have often been regarded as dispensable and have been skipped in case of tight schedules [19].

- Existing HCD process models are not suitable for tailoring.

Another point that is also ignored by the approaches described is that development organizations are often overwhelmed by the sheer complexity of the proposed HCD process models. The models lack a defined procedure for tailoring the development process and methods for specific project constraints as system domain, team size, experience of the development team or the system development process already practiced by the organization.

- Existing HCD process models assume that HCD methods can be performed ad hoc.

Most approaches also assume that experienced human factors specialists are available throughout the development team and that HCD methods can be performed ad hoc. However, recent research shows that even highly interactive systems are often developed without the help of in-house human factors specialists or external usability consultants [20]. Therefore HCD methods often can not be utilized because the necessary knowledge is not available within the development teams [19, 21].

- Existing HCD process models are not prepared for continuous process improvement.

Almost all approaches do not account for the fact that turning technology-centered development processes into human-centered development processes must be seen as a continuous process improvement task [4]. A strategy for supporting a long-lasting establishment of HCD knowledge, methods, and tools within development organizations is still missing.

### 3 The Experience-Based Human-Centered Design Lifecycle

The experience-based human-centered design lifecycle addresses the shortcomings of the existing HCD process models described above. It aims at promoting the utilization of HCD activities and methods in development processes practiced by software development organizations. It supports the introduction, establishment, and continuous improvement of HCD activities, methods, and tools within software development processes. Our approach is based on two concepts: An *HCD reference model* that contains specific activities for each logical phase of the development process and the *introduction-establishment-improvement (IEI)* model for HCD activities which guides the tailoring and smooth integration of HCD methods in mainstream software development processes.

#### 3.1 The HCD Reference Model

The reference model contains HCD activities to be integrated into software development processes for interactive systems. It is organized into virtual process phases based on the framework of the *usability maturity model* (UMM) defined in ISO TR 18529 [22] which is based on ISO 13407 [17] and the Capability Maturity Model (CMM) [33]. The reference model contains various HCD activities from different human-centered design approaches. Currently the model comprises activities adopted from Nielsen [6] and Mayhew [10] as depicted in Fig.1. The shaded blocks represent the UMM framework while the plain blocks represent the respective activities of our reference model<sup>1</sup>.

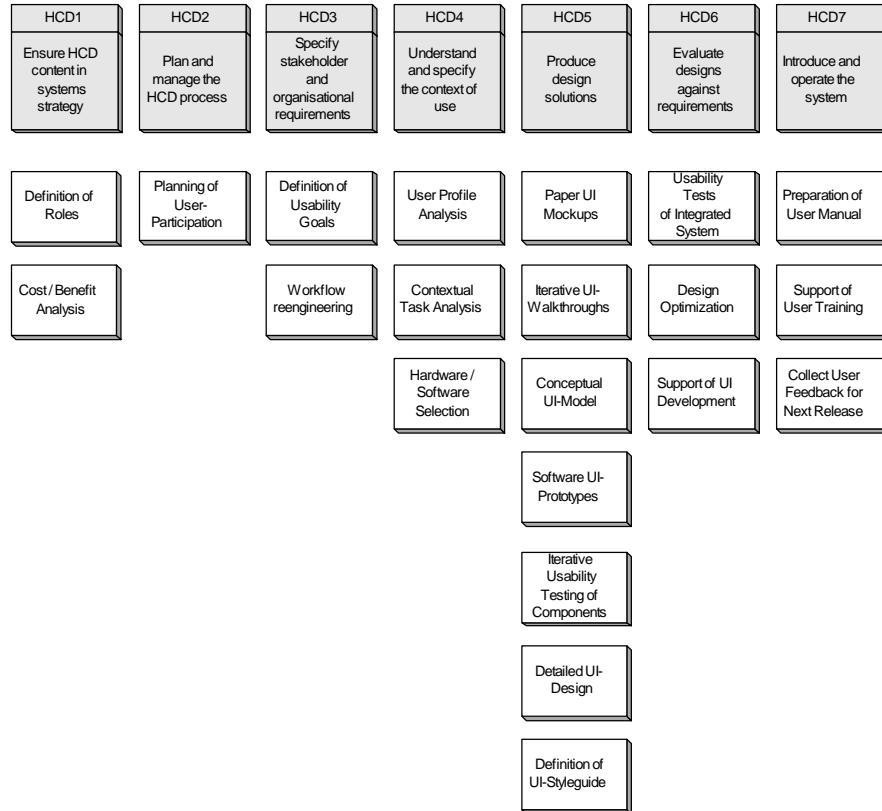
The set of methods collected in the reference model is continually revised and extended with new methods by using the model in real projects and collecting data on its efficiency. The aim of the reference model is to compile a collection of well-accepted HCD activities in one framework which have been carefully verified for practicability and efficiency together with context information (process context, project context, technology context, domain context, and quality context) that describes in which development context the respective activities are most efficient.

#### 3.2 The Introduction-Establishment-Improvement Model for HCD

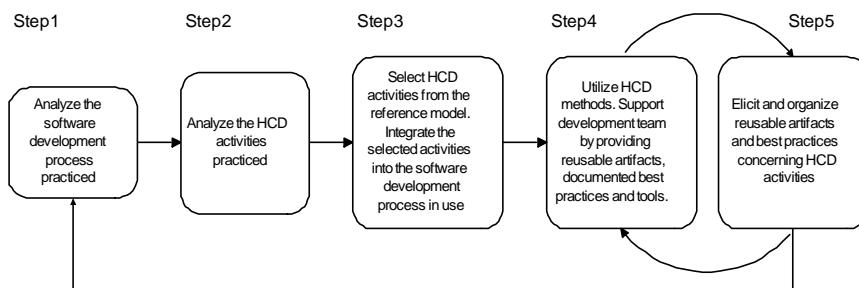
The set of activities of the reference model must be supplemented with organizational tasks which help to manage and tailor the HCD activities of the reference model according to specific constraints of the respective project and the needs of the development organization. Our introduction-establishment-improvement (IEI) model for human-centered design provides process steps that support the introduction,

---

<sup>1</sup>The graphical representation of the reference model is adopted from the format of ISOTR18529 – ‘Human-centred lifecycle process descriptions’. Whilst it is possible to draw a number of simple diagrams which demonstrate the iterative nature of the human- centred lifecycle there are many different versions of lifecycles, depending on the development context of the system planned. It is therefore difficult and may even be confusing to draw one simple diagram which demonstrates how processes are linked.

**Fig. 1.** Mapping of the Reference Model to the UMM framework [22]

establishment and continuous improvement of HCD activities throughout the whole development lifecycle of highly interactive systems. These organizational tasks are grouped in the IEI model as depicted in Fig. 2.

**Fig. 2.** The IEI model

The IEI model consists of the following steps:

*Step 1: Analyze the software development process practiced*

At the beginning, a short analysis of the software lifecycle practiced is performed. The results of this step are, first, a rough description of the process model that is used by the software development organization, and, second, a rough description of its main phases.

*Step 2: Analyze the HCD activities practiced*

The next step comprises an analysis of the practiced HCD process and the related HCD activities, to elicit when, where and how HCD methods are performed within the software development lifecycle in use. The deliverable of this step is a documentation of possible improvements of the HCD process that is currently used.

*Step 3: Select suitable HCD tasks and integrate them into the practiced software development process*

The results of the first steps form the rationale for the selection of HCD activities from the reference model for the improvement of the development process. However, in this step further important factors have to be considered, e.g. the type of system to be developed and project constraints like budget and schedules.

The HCD activities which have been selected for the improvement of development process have to be integrated in the model of the practiced software development lifecycle and the project planning and form the improved development process, supplemented with appropriate HCD activities.

*Step 4: Support effective performance of the defined HCD activities*

Generally, at this step in the model resources have already been allocated for HCD activities, e.g., a usability engineer was nominated, who is responsible for coordinating and supporting the execution of the various HCD activities of the new process. However, the efficiency and impact of the proposed HCD methods must be increased by providing information, best practices, tools and reusable process deliverables of past projects (e.g. templates for usability test questionnaires, results of conceptual task analysis or user interface mockups) which facilitate effective performance of the selected HCD activities. This set of information should be easily accessible for all participants of HCD activities.

*Step 5: Collect and disseminate best practices and artifacts concerning HCD tasks*

During the execution of HCD activities, artifacts with a high value for reuse in the same or subsequent projects are generated by the participants of HCD activities, for example, templates for usability tests, reusable code fragments, or best practices on how to most efficiently conduct a user profile analysis. Furthermore observations of gaps in the HCD process should be documented and used as an input for the next improvement cycle. These best practices and artifacts comprise HCD experience and rationale that have to be captured and organized in a way that allows for easy reuse in the same or subsequent projects.

The IEI model contains two cycles: The inner cycle between step 4 and 5 supports the introduction and establishment of HCD activities and methods within the practiced software development process. It supports the effective utilization and improvement of HCD activities and best practices which are tailored to the needs of the development organization. This cycle is continuously iterated during the development process.

The outer cycle which connects step 5 and 1 should be performed in the ideal case at least twice during the development process of large projects as it serves the improvement of the overall HCD process. In this cycle observations and best practices collected in step 5 should be used to integrate needed HCD activities in the process and remove activities that have failed to prove their utility.

## 4 Tool Support

The literature often maintains that there is a lack of tool support for HCD processes [6, 10, 19, 21, 23]. Indeed, few mature tools are commercially available most of them for activities of the late phases of the development lifecycle, e.g. (semi-) automatic GUI (graphical user interface) builders [24-28] or usability test tools [23, 29, 30]. With the increasing relevance of usability as a software quality factor, there will be linked a growing demand for tools which support HCD activities within the industrial software development community. These tools should help the participants of the HCD lifecycle to execute their tasks more effectively.

One important class of HCD tools are systems that support developers by providing the knowledge to perform the various HCD activities and to organize the outputs of iterative design processes [10, 19, 21]. For the lack of available commercial tools, we developed a prototypical tool, the so-called *MMI (Man-Machine-Interaction) hyperbase*. The development and introduction of the system was prepared by a requirements analysis conducted with eighteen professional software developers and six project team leaders [31].

The tool summarizes and presents ergonomic design knowledge and HCD methods for the entire HCD lifecycle using a knowledge base and makes the knowledge available via intranet and web-browser. The ergonomic design knowledge and the HCD methods have been attached to the different HCD activities of our reference model, e.g. ‘definition of usability goals’ or ‘user profile analysis’. The MMI hyperbase supports a software development organization in defining the roles of the project team members responsible for HCD and helps to select the right methods and tools for each development step. Furthermore the MMI hyperbase supports the cooperation between the different participants of the HCD lifecycle with the help of a groupware tool. It enables developers to bundle best practices learned during the performance of HCD activities together with reusable artifacts and attach them to the corresponding activities of the reference model. The aim is to reuse HCD rationale, support organizational learning concerning HCD and to improve the HCD process deployed, and thus supports our experience-based approach outlined in the IEI model.

## 5 Employment of the Experience-Based HCD Lifecycle in an Industrial Setting

In the remainder of this paper we illustrate the application of the methods and tools described above on the basis of a military software development project at DASA<sup>2</sup>, called AP3In this project we have supported the introduction, establishment and improvement of HCD activities within the software development lifecycle. The overall aim of our activities in AP3 was to increase the productivity of the user interface development process and to improve the quality-in-use of the system developed. In the following section we briefly describe how each step of the IEI model was applied in AP3:

*Step 1: Analyze the practiced software development process*

The analysis of the practiced software development lifecycle to be used in AP3 showed that the development department used a subset of the V-model [32] as a process model for the development of their software systems. The V-model describes the activities which have to be performed during system development and the deliverables - mainly documents - which are to be created during these activities. The software process model to be used by the development team comprised the following five logical phases: project preparation, requirements analysis, system design, system development and installation/transition into use.

*Step 2: Analyze the practiced HCD process*

The analysis of the V-model with respect to HCD activities showed that there is no explicit HCD process anchored as a sub-process in the V-model. Therefore, HCD activities and their relation to the system development phases had to be defined for all phases of the development process.

*Step 3: Select suitable HCD tasks and integrate them into the practiced software development process*

Based on the results of the first two steps, the type of product to be developed and the project constraints, a set of appropriate HCD activities was selected from the reference model. Some further constraints influenced the selection of HCD activities: Though an old system existed, totally new work flows were to be modeled for the new system, thus making it necessary to integrate the workflow reengineering activity into the development process. Furthermore the proposed system had to comply with legal regulations concerning usability [1], which had to be taken into account in the process activities *definition of usability goals* and *usability testing*.

The HCD activities which have been selected from the reference model were integrated into the respective phases of the practiced development process as elicited in step 1 and formed the process model for the performance of HCD activities in the AP3 project.

The resulting process model is depicted in Fig. 3. The process phases of the overall development process which have been elicited in step 1 are represented as shaded

---

<sup>2</sup>DaimlerChrysler Aerospace AG (DASA) is now partner of the transnational European Aeronautic Defence and Space Company (EADS)

boxes. The plain boxes represent the HCD activities that have been selected from the reference model to supplement the development process. The phases must be understood as logical phases. This means that they do not have to be performed in strict order for the whole system like in the waterfall model, but that they can be iteratively applied to sub-components. However, for reasons of clarity and readability we forgo the drawing of iterative loops between the logical process steps that occur in every real project. The resulting process model served as a basis for allocating human resources and project schedules in AP3.

*Step 4: Support effective performance of the defined HCD activities*

The setup and execution of the HCD activities was coordinated by a usability engineer who was released from other project activities. The impact of the usability engineer was enhanced by the MMI hyperbase tool and the knowledge base included. With this intranet-based tool, the developers had access to practical descriptions of the various tasks of the HCD process set up in step 3, as well as to auxiliary means, e.g. templates, checklists, existing components or documented best practices, at all times - which made the efficient execution of the usability activities possible.

*Step 5: Collect and disseminate best practices and artifacts concerning HCD activities*

During the execution of the various HCD activities, some promising best practices were created. For example streamlined methods for user profile analysis and contextual task analysis together with checklists were developed to facilitate efficient performance of respective HCD activities.

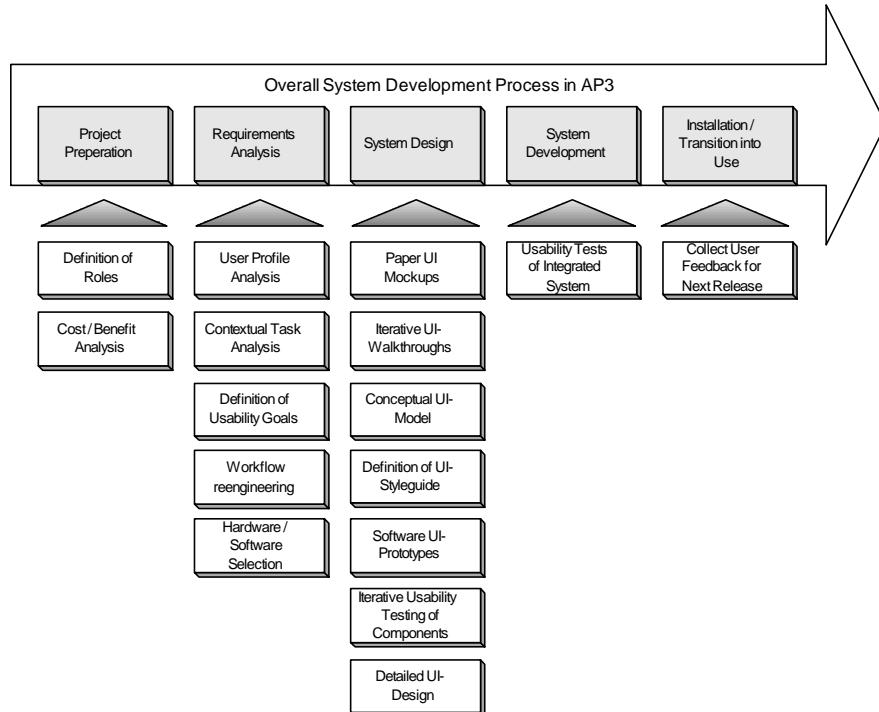
Furthermore, a simple yet effective graphical notation for user-relevant system functionality and work flows was developed during the workflow reengineering sessions. Based on use cases and flow diagrams this notation was both understood and accepted by the contractors, users and software developers thus supporting the communication during the workflow reengineering activity.

These best practices have been documented in the knowledge base of the MMI hyperbase together with information like the HCD activity to which they are related, the role for which they are most useful (e.g. usability engineer, user interface (UI) designer, UI developer).

Based on the experiences collected in the inner cycle of the model, by iterating steps 4 and 5, we went back to step 2 to improve the practiced HCD process. The improved HCD process was used in the remainder of the AP3 project.

## 6 Lessons Learned

The best methods and tools for developing usable systems cannot achieve full impact if they are not supported by suitable organizational measures. Here are some of the simple measures which contributed to the successful application of usability engineering methods in the AP3 project:



**Fig. 3.** Integration of HCD activities in the development process of AP3

- Nominate personnel which is responsible for HCD during the whole project  
For the successful introduction and establishment of HCD activities it proved to be mandatory to have usability experts involved in both the development organization and the customer organization. In the development organization a usability engineer should be involved right from the beginning of the project and should be responsible for the management of all HCD activities during the whole system development process. In the AP3 project for example the usability engineer was explicitly declared as a member of the core engineering team and was also involved for the requirements engineering phase. On the customer side a usability agent was supervising the system's compliance with usability requirements of the customer organization. In AP3 one usability engineer and one usability agent were nominated who were intensively cooperating.

- Perform regular usability audits with stakeholders

From our point of view HCD is also always a process of negotiation between different stakeholders, e.g. users, contractor, developers, management and goals of one group often clash with the interests of others. To support communication and facilitate a maximum degree of agreement between these groups on the most important design decisions, usability audits were performed throughout the development process. Each of this audits required at least two days of collaborative work and should be carefully prepared including the needed documents and tools.

During the AP3 project a total number of five usability audits were performed within three years.

- Support the development team with information

One of the most important recommendation is to compile all relevant information and data which is produced during the HCD activities and make it available to all members of the development team. The important results of the early phases such as user profiles, user tasks, or usability goals have to be considered during the whole development process and are vital for the late HCD activities such as testing for compliance with usability goals, designing the user manual by describing major user tasks or supporting user training by explaining and practicing the user tasks at the target system. It has proved to be useful to make the deliverables of the performed usability engineering tasks available in a central repository to all participants of the usability engineering process. In AP3 we used the MMI hyperbase as an HCD support tool.

## 7 Results

There are some important findings to our HCD approach. In comparison to other similar projects in our organization during the last years we identified a lot of different benefits for the AP3 project which was driven with our approach:

### 7.1 General Advantages and Savings

In contrast to previous projects the interaction design and user interfaces were available very early in AP3. Both sides got an early impression of the system's look and feel. The customers felt that they get a system which corresponds to the needs of their users. The suppliers got the certainty that they will deliver a system which satisfies the users. In AP3 the final electronic user interfaces were completed after 60% of the overall development time. In former projects with the same settings usability problems resulted in massive exceeding of deadlines.

### 7.2 Advantages Related to the Experience-Based HCD Lifecycle

One of the most valuable advantages for the development team was that developers and software engineers learned to think in categories of user tasks and not only in terms of system functionality.

In former projects developers only got lists of functions and often did not understand the defects which were communicated to them by the customer. This resulted in user interfaces that rather depicted a collection of single functions than a representation of the users tasks. It was annoying for developers to follow the change orders of the customer.

From the very beginning of the AP3 project the software developers got a deep insight into the user tasks since they participated in performing the contextual task analysis together with the user representatives. Due to this experience they were highly motivated to implement components with a high usability. They were able to

understand and solve the problems encountered during usability tests since they knew the user tasks and they realized a mismatch between those tasks and the prototype. Furthermore the whole AP3 project profited from the elicitation and reuse of best practices and artifacts as described above. The development team adapted known HCD activities and methods and streamlined them to fit the development process in used, thus continuously improving the HCD activities practiced.

### 7.3 Synergy Effects Related to Tool Support for HCD Processes

There were immense cost savings since a lot of the material which was produced during the HCD process could be reused after delivery of the system in many ways. One example was the user training material. After delivery the customer organization normally defines an own training team which is responsible for the training of new users. This team got an excellent training material including all the descriptions of user tasks, usability goals, style guide rules, and tests cases for the user interfaces that was compiled using the MMI hyperbase tool. The effort for the training of new users was clearly reduced.

## 8 Conclusions

We presented an experience-based approach to HCD which was designed to overcome the severe organizational obstacles encountered during the application of related approaches. With the help of a case study we showed how to employ our approach in industrial software development organizations and presented organizational measures which contributed to the success of the entire approach. We presented promising results of the case study that showed that applying our approach can facilitate a more effective utilization of HCD methods within industrial software development settings and support the introduction, establishment and continuous improvement of HCD activities in software development organizations.

## References

1. EC Directive 90/270/EWG: Human-Computer Interfaces, Directive about Safety and Health of VDU-Work (5th Directive Art. 16 Par. 1 of Dir. 89/391/EWG). Official EC-Newsletter, Vol.33, L 156, Minimal Standards, Par.3 (21.6.1990) 18., 1990.
2. Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten, Bildschirmarbeitsverordnung - BildscharbV, BGBl. I 1996 pp. 1841, 1996.
3. J. Nielsen, *Designing Web Usability : The Practice of Simplicity*: New Riders Publishing, 2000.
4. D. A. Norman, »Chapter 10: Want Human-Centered Design? Reorganize the Company.,» in *The Invisible Computer*: MIT Press, 1998.
5. J. D. Gould and C. Lewis, »Designing for Usability: Key Principles and What Designers Think,» *Communications of the ACM*, vol. 28, pp. 360-411, 1985.
6. J. Nielsen, *Usability Engineering*: Morgan Kaufman Publishers, 1994.

7. J. Karat and T. Dayton, »Practical Education for Improving Software Usability,” presented at Conference on Human Factors in Computing Systems: CHI'95, 1995.
8. D.R. Wixon J. Ramey (eds.), Field Methods Casebook for Software Design. New York: John Wiley & Sons, 1996.
9. H. Beyer and K. Holtzblatt, Contextual Design: Defining Customer-Centered Systems: Morgan Kaufmann, 1998.
10. D. J. Mayhew, The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design: Morgan Kaufman, 1999.
11. L. L. Constantine and L. A. D. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design: Addison-Wesley, 1999.
12. W. W. Royce, »Managing the development of large software systems,” presented at IEEE WESTCON, San Francisco, USA, 1970.
13. B. W. Boehm, »A spiral model of software development and enhancement.,” IEEE Computer, vol. 21, pp. 61-72, 1988.
14. B. Henderson-Sellers and J. M. Edwards, »Object-Oriented Systems Life Cycle,” Communications of the ACM, vol. 31, pp. 143-159, 1990.
15. P. B. Checkland, Systems Thinking, Systems Practice: John Wiles & Sons, 1981.
16. D. Hix and H. R. Hartson, »Iterative, Evaluation-Centered User Interaction Development,” in Developing User Interfaces: Ensuring Usability Through Product & Process. New York: John Wiley & Sons, 1993, pp. 95-116.
17. ISO/TC 159 Ergonomics, »Human-centered Design Processes for Interactive Systems,” ISO International Organization for Standardization ISO 13407:1999(E), 1999.
18. P. B. Checkland and J. Scholes, Soft Systems Methodology in Action: John Wiley & Sons, 1990.
19. S. Rosenbaum et al., »What makes Strategic Usability Fail? Lessons Learned From the Field.,” presented at International Conference on Human Factors in Computing Systems, CHI99, Pittsburgh, PA USA, 1999.
20. E. Wetzenstein, A. Becker, R. Oed, »Which Support do Developers Demand During the Development of User Interfaces” (in german), to be presented at the 1<sup>st</sup> interdisciplinary Conference on Human-Computer Issues, M&C2001, Bad Honnef, Germany, 2001.
21. H. Reiterer, »Tools for Working with Guidelines in Different User Interface Design Approaches,” presented at the Annual Workshop of the Special Interest Group on Tools for Working with Guidelines, Biarritz, France, 2000.
22. ISO/TC 159 Ergonomics, »ISO TR 18529 - The Usability Maturity Model.”
23. G. Al-Quaimari and D. McRostie, »KALDI: A Computer-Aided Usability Engineering Tool for Supporting Testing and Analysis of Human-Computer Interaction,” presented at CADUI99: Computer Aided Design of User Interfaces, Louvain-la-Neuve Belgium, 1999.
24. Foley et al., »UIDE - An Intelligent User Interface Design Environment,” Intelligent User Interfaces, pp. 339-384, 1991.
25. P. Szekley et. al., »Facilitating the Exploration of Interface Design Alternatives: The HUMANOID Model of Interface Design,” presented at Conference on Human Factors in Computing Systems, CHI'92, 1992.
26. Balzert et.al., »The JANUS Application Development Environment - Generating more than the User Interface,” presented at Computer Aided Design of User Interfaces, CADUI'96, 1996.
27. A.R.Puerta, »A Model Based Interface Development Environment,” IEEE Software, vol. 14, pp. 41-47, 1997.
28. J. Vanderdonckt, »SEGUIA - Assisting Designers in Developing Interactive Business Oriented Applications,” presented at internatl. Conference on Human Computer Interaction, HCI'99, Munich, 1999.
29. M. Macleod and R. Rengger, »The Development of DRUM: A Software Tool for Videoassisted Usability Evaluation,” presented at BCS Conference on People and Computers VIII HCI'93, Loughborough, 1993.

30. D. Uehling and K. Wolf, »User Action Graphing Effort (UsAGE),” presented at ACM Conference on Human Aspects in Computing Systems CHI'95, Denver, 1995.
31. H. Wandke, A. Dubrowsky, and J. Huettner, »Anforderungsanalyse zur Einführung eines Unterstützungssystems bei Software-Entwicklern,” presented at Software Ergonomie '99, Walldorf, Germany, 1999.
32. IABG, »Vorgehensmodell zur Planung und Durchführung von IT-Vorhaben - Entwicklungsstandard für IT-Systeme des Bundes,” : IABG- Industrieanlagen Betriebsgesellschaft, <http://www.v-modell.iabg.de/>.
33. Mark C. Pault, Bill Curtis, Mary Beth Chrissis, and Charles V. Weber, "Capability Maturity Model, Version 1.1," IEEE Software, Vol. 10, No. 4, July 1993, pp. 18-27.

## Discussion

*L. Bass:* How many best practices were incorporated in the project?

*E. Metzker:* Around 30.

*G. Phillips:* How do you control what information regarding best practices make it into the knowledge base?

*E. Metzker:* The project manager and the responsible user interface engineer determined what was stored in the knowledge base.

*F. Paterno:* How did you evaluate the success of the approach in your case study and what elements contributed to it?

*E. Metzker:* We need more case studies to better understand the elements that are more relevant and how they are useful.

*H. Stiegler:* You talked a lot about the UMM which is strongly related to CMM. From my experience it is hard work to arrive at a level beyond 3. Your tool approach even deals with the improvement cycles between level 4 and 5. Did you really achieve that level in such a short time frame?

*E. Metzker:* We did not make a formal assessment yet and we do not claim to have achieved these levels but our overall planning includes the goals of continuous process improvements.