

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221610493>

Integrating User Centered Design in a Product Development Lifecycle Process: A Case Study.

Conference Paper · January 2006

Source: DBLP

CITATION

1

READS

695

3 authors, including:



Karsten Nebe

Hochschule Rhein-Waal

59 PUBLICATIONS 335 CITATIONS

[SEE PROFILE](#)



Ronald Hartwig

rhaug GmbH

19 PUBLICATIONS 52 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Trust in Automation [View project](#)

Integrating User Centered Design in a Product Development Lifecycle Process: A Case Study

Karsten Nebe
C-LAB
University of Paderborn
Paderborn, Germany

Karsten.Nebe@c-lab.de

Lennart Grötzbach
C-LAB
Siemens Business Services
Paderborn, Germany

Lennart.Groetzbach@c-lab.de

Ronald Hartwig
C-LAB
Siemens Business Services
Paderborn, Germany

Ronald.Hartwig@c-lab.de

The authors will present their experiences gathered by implementing a User Centered Design (UCD) Process in an existing product development lifecycle within the complex setting of a large healthcare software company. The company started a project to define the next release of a software solution for the physician's workplace using UCD methodology. The project goals were twofold: First to develop a front-end integration of various IT-systems of a physician's workspace that meets user expectations and improves their workflows, and second to prove that the UCD methodology creates substantial advantages for the organization.

Keywords: User Centered Design Process, Usability Engineering, Software Engineering, Requirement Engineering, Use Case

1. Introduction

The methodology of User Centered Design (UCD) is well established and described in today's literature ([5], [6]). From the authors' experience it can be said that even though the methodology is well known many companies still do have problems to adopt UCD techniques and methods to their product development lifecycle. They recognize the need to increase the usability of their products but fail to apply it.

This case study will highlight the authors' practical experiences and findings they gathered during the implementation of an UCD process in an existing product development lifecycle of a large healthcare software development company.

The company's products address a diversified market in the diagnostic healthcare sector, e.g. general hospitals, university hospitals and practices in the USA and Europe. Even if their products are well accepted and widely used by different customers (in terms of applying organizations) the company realized that the future success of the software products would require a change towards a more user and workflow

centered design approach as opposed to today's technology driven practice. Implementing features and functions based on non-validated (often singular) customer requirements resulted in unnecessary costs. The lack of a defined way of documenting and using user and related usability requirements made products inconsistent and beyond this, made applying product changes difficult. If mismatches between expected and real users were discovered later, this lead to further costs (e.g. maintenance) [1] and in the meantime many customers had to deal with the features they did not really need.

To explore how an UCD process can support the development process and create more usable products, a pilot project has been started to define the next software release of a physician's workspace located in a medium sized hospital. The new software release covers special areas of the daily work of the physicians, but due to confidentiality reasons the paper cannot dwell into specific project details.

Two main goals have been defined for the project: First, to develop a front-end integration and unification of different existing software solutions and second, to improve the users' workflow support. As an organizational meta goal the project had to prove whether an UCD process can create substantial advantages for an organization or not.

If the project proved to be successful the company plans to change their entire product development lifecycle for this product sector. In order to reach these goals, several organizational changes have been applied within the company. In addition the UCD activities have been aligned with the product development lifecycle, and especially for the requirement engineering process.

The basis of UCD is the knowledge about the users and their tasks. Based on this it is possible to develop integrated solutions that meet the users' needs and fit to their workflows. However, the knowledge of the types and relevance of customers has also to be considered to create a substantial advantage for the

user and the customers' organizations. Thus, the company needs to understand the users and the market likewise.

Altogether, it has to be proven that the introduction of the UCD process has succeeded in creating the new release of the physician's workspace that is more usable and supports the users' workflows more effectively.

2. Proceedings

To implement the UCD process within the organization, the authors analyzed the given product development lifecycle and identified the process steps to be adopted, replaced or added, according to existing process models ([3], [7]). The authors found it adequate that existing activities in software development and requirement engineering had to be considered because in a day-to-day business they cannot be reinvented from scratch. In order to achieve the project goals, the authors first determined the major challenges and opportunities for the project in cooperation with the marketing and innovation department. This laid the groundwork and established a common understanding of the project challenges surfaced potentials for organizational changes and process adjustments.

To counter resistance it is necessary to create a common understanding early and to point out the advantages and benefits of UCD for all parties involved. It is important to be aware of "hidden" extra efforts and new work packages which may impede the acceptance of this method within existing teams. To overcome these reservations the possible benefits must be communicated from the beginning on.

The authors established an interdisciplinary process team with members from different educational backgrounds and belonging to different departments (marketing, development, system architecture, innovation management, testing, project management, product analysis and usability engineering) in the organization.

A new role that was created within this project is the Product Analyst (PA). PAs are domain experts, such as medical doctors, technical assistants or unit clerks with long time experience in their field of work. Having domain experts in the process team is crucial because they can incorporate their knowledge about the users' workflow and the organizational settings the users operate in. Another advantage is that these experts are, unlike real users, always accessible to the team, e.g. to answer questions. PAs protect the end users interests during the project and gather end users' information and have the ability to structure and revise this information in an adequate way. With PAs at hand end-user involvement can be restricted to analysis and validation activities which is also time and cost saving.

The authors also introduced the Concept Team which is responsible for translating analysis results into concepts and subsequently into user interface designs. The concept team consists of designers and usability engineers who work hand in hand with the PAs and the development department.

The interdisciplinary set-up of team members and their close collaboration among them enables a rapid exchange of knowledge and information that also provide a basis for continuously inspections of the results.

2.1. Accomplishment

UCD activities need to be aligned with the existing product development lifecycle and new tasks, roles, artifacts (documents) and interfaces between the existing process and the new activities need to be defined. The interplay of the documents with the documentation and requirement engineering process has to be defined soundly and traceable.



Figure 1. Integrated model of the User Centered Design Process and the Product Development Lifecycle

The UCD process model as applied in this project consists of four phases (Figure 1). Each phase must be seen as a self-contained iterative process which can be performed from an abstract to a detailed level. All phases overlap and run in parallel - in particular the validation phase which consists of several activities that are geared towards all other process phases interacts with all other phases.

The Analysis Phase consists of several levels of analyses. First, the target market has to be analyzed in order to understand the customer's workflows, followed by an analysis of the users of the future system. The knowledge about the users is essential to create stakeholder-based solutions. Finally, the user's tasks have to be analyzed to get a detailed understanding about the users' workflows and their respective goals. The Collaboration and Conceptualization Phase connects the knowledge of the customers' workflows, the users of the system, and how the users perform their daily work, which results in concepts and designs that meet the corresponding user and workflow requirements. These concepts are implemented during the

Implementation Phase. The developed solutions are adapted in the Validation Phase until they meet the users' expectations and suit their needs and fit their workflows.

2.1.1. Business process level analysis. To get a better understanding of the customers' workflows and to identify the differences and associated business requirements, the marketing experts conducted a detailed market analysis. The marketing experts analyzed sales characteristics, customer feedback, competitors' products, performed semi-structured interviews, conducted site visits at different customers' sites and documented their findings in Business Use Cases and Business Requirements.

They derived an abstract workflow schema by clustering the sub-tasks which allows them to compare the workflows in relevant contexts of use with each other. As a result, it turned out that the workflows at the different customer's sites are almost the same despite the variety of organizational contexts. Noticeable differences could only be found in the execution of single tasks but nevertheless the differences do not influence the overall sequence of workflow steps. As a consequence, an overall workflow was identified called the "Meta Workflow".

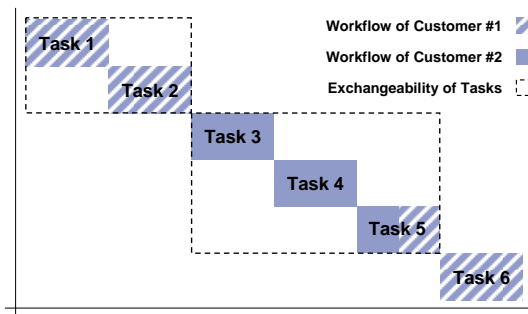


Figure 2. Exemplarily figure of the Meta Workflow and interplay of two customers' Meta Tasks

The sequence of the Meta Workflow steps build up an order of Meta Tasks that represent the flow of work at the customers' sites (Figure 2). Each Meta Task encapsulates a number of sub-tasks that have to be performed by the users. The number of Meta Tasks handled by a customer depends on the organizational setting. For example, a university hospital will more likely handle most of the Meta Tasks on their own, in contrast to a practice that focuses only on their specialty in terms of specific Meta Tasks. Each Meta Task has to be completed, before the next Meta Task can start. Thus, the Meta Tasks are interrelated.

Within the Meta Tasks, results vary by the type of organization, e.g. the prioritization of certain activities, the automation of certain tasks or the replacement of tasks by integration of third party products. Different users have to deal with certain Meta Tasks cooperatively to ensure the flow of work

according to the Meta Workflow. Hence, there is a given sequence of Meta Tasks for each user. To offer stakeholder based solutions, these sequences of users' Meta Tasks need to be represented in the product portfolio and finally in the future software solution, of course.

The Meta Workflow proved to be very valuable, especially for the marketing and the development department. Previously, the company implemented several product lines for different market segments (types of customers). All these product lines had to be developed, coordinated, and maintained. The Meta Workflow enables the company to define just one product line and adapt any sequence of Meta Tasks to the different customers' settings. Finding and identifying broad similarities helped to cut down development efforts and to concentrate on significant workflows. Since the Meta Tasks are encapsulated and the variations are bounded by the Meta Task, each Meta Task could be implemented as a single Task Module fitting the overall Meta Workflow. All variations can be addressed within a Task Module and as a result the modules fit together like a construction kit.

Having identified the Meta Workflow the authors continued to get a detailed understanding of the users and their tasks. To achieve this further analyses have been conducted at different customer sites.

2.1.2. User level analysis. The detailed analysis of the users, their workflows and tasks was conducted according to established UCD methods as described by Beyer & Holzblatt [2], Mayhew [7], and Nielsen [8]. As defined within the project scope, the daily business of the Physician and the Technological Assistant in the different organizational contexts have been analyzed in more detail, summarized and documented in a so called "Persona Use Case". To get a better understanding of the nature and the interrelationship of the user's daily work, the analysts performed a high-level structured task analysis and documented the findings in the persona use case as well. The outcome is a high-level overview of all the users' tasks, categorized by frequency, duration, character, priority, etc.

A persona use case, as used in this project, is an abstract overview about a specific group of users. It is a conglomeration of the analysis results of multiple users in various contexts. The persona use case consists of a description of the user group and the variations depending on the educational status and their context of work (e.g. physician in practice, head physician in a hospital or practice). It also includes impacts of the work environment, regional distinctions, and legal policies. Thereby, the persona use case gives a holistic image of one potential group of user of the future system, their work environment, their daily business and their daily tasks. Its main

purpose is to get a defined and stable project wide common understanding of the users, their job and their environment. The narrative description of the users' work is an easy and intuitive way to communicate this knowledge through out the whole organization. It can be read and understood by anyone in the organization in contrast to many formal methods.

The structured task analysis as an additional part of this document provides an abstract high-level overview of the goals and tasks that are in the scope of the software solution. Based on this, the personas (and their tasks) can be mapped with the Meta Workflow and the Meta Tasks. If there is more than one persona assigned for one Meta Task, primary and secondary roles (in terms of personas) are identified. This helps to prioritize tasks and roles and their corresponding requirements later on. From the users' and a workflow perspective it is mandatory to consider always the complete workflow at the customers' site to ensure a continuous solution. If such a workflow involves more than one persona, other mandatory roles have to be considered here (e.g. if physician and assistant are both needed to complete a task).

Based on the mapping of the personas' tasks, according to the Meta Workflow, the Meta Tasks have been analyzed in further detail.

2.1.3. Task level analysis. Each Meta Task of the Meta Workflow is defined as a "User Goal Use Case" (similar to the definition of Cockburn [4]) to be analyzed and documented in higher detail. A user goal use case reflects the fundamental goal of a user, performing a specific task. To gather detailed information about how the users work, the product analysts performed site visits, conducted interviews, and documented the findings in the corresponding user goal use case documents.

The User Goal Use Case, as applied in this project, describes the flow of work from a user's perspective. It reflects the real life situation of the user and how the user has to deal with certain situations while performing a specific task. The use case does not include a description of a solution, but the description of the mandatory sub-tasks and decisions that have to be performed in order to complete the user's goal. It includes all information and data the user needs, a description of tools the user works with and also the variations and alternatives to his routine. The detailed flow of work is documented in so-called "Use Scenarios" which represent the intention and the corresponding expected system response for each single step in the user's workflow.

There is always a single "Main Scenario" that represents the standard flow of work. Any variation or branches out of the main flow are documented in

"Alternative Scenarios". Use scenarios are always complete workflows, not just single actions or events.

User goal use cases provide highly detailed information about the way the users work. There are no technical descriptions of features and functions. They contain no information about a possible or existing solution, but detailed description about the user's intent and the user's expectations towards the system.

This abstraction on intentions and goals brings the user in the position to give more detail about his needs instead of what he expects, which is normally too close to an existing solution. Pairs of intentions ("What do I want here?") and needed system responses ("What would I need to get there?") form use scenarios which are still human readable. This format aids validation efforts for all stakeholders, involved in the development process later on.

The scenarios deliver user requirements for the context of use, which are being defined during the implementation phase. Having understood the user needs and tasks is an inevitable precondition to adhere to the usability factors of effectiveness, efficiency and satisfaction. This is the basis for all further development process steps.

Within an early iteration phase of the process, the analysis was conducted only focusing on the primary persona. Personas that are supporting or assisting a specific Meta Task have been analyzed later on. Focusing on the key-persona helps to separate the important tasks from the less important tasks while still focusing on the users' workflows. Thus, the primary persona also supports the decision making process of a scope definition.

The use case documents capture the knowledge about the users, their tasks and the organizational contexts and works as a common knowledge base for further elaboration among analysts, development, management, etc.

2.2. Collaboration & Conceptualization

The use cases are now used as a communication medium across departments involved. They foster a common understanding of the product scope. The persona use cases help to understand the users' daily business and environment. The user goal use cases help to get a deeper understanding of the users' goals and how they perform domain specific tasks. With this knowledge the concepts are created.

As a new activity in the process of the organization the authors introduced the concept of collaborative sessions to discuss and clarify process outcomes and the corresponding documentations. Depending on the progress and the detail level of documentation different parties of the organization are involved:

- The PAs, the marketing department and product management commonly work out the product portfolio and discuss management issues to place

stakeholder-based solutions for the market, based on the Meta Workflow and the persona use cases.

- PAs, system architects and the development department clarify technical feasibilities and design details, based on the use scenarios of the user goal use cases. As a result, the use cases are refined and additional technical specifications are gathered and documented in so called “Supplemental Use Case Specifications”. The sum of all these documents can be seen as the foundation for the development of concepts for the users’ tasks.
- As soon as the documents have reached a certain level of maturity, the concept team begins to create first high level concepts. Then, collaboration sessions between PAs and the concept team are being used to validate them.
- In a follow-up session with system architects and the development team the technical feasibilities are discussed.

The creation of concepts follows a defined way of translating the use scenarios into solutions that fit the users’ flow of work. The precise documentation of users’ intention and their’ expected system response (encapsulated in the use scenarios) is used to create concepts. Thus, the concepts are linked to the use cases and to the corresponding use scenarios.

Once the involved teams agree to a concept, the creation of the final designs can start. The concepts and user interface designs are documented in so called “UI Solution Paper”, user interface specification documents that describe the design of the solution and the interaction within the solution.

Collaboration sessions take place until the teams reach an agreement about the open issues. The outcomes of each session are documented. The overall goal of the collaboration sessions is to create a common understanding and a mutual agreement on all working materials.

Use cases were an adequate communication medium, for a wide variety of people within the different departments. The teams work commonly together on the same level of user information

2.3. Implementation

The company uses a component-based software engineering approach for developing their software. The reuse of components turned out to be the appropriate procedure for building software solutions in such a diversified market. One organizational constraint was that this underlying model must not be changed. To meet this requirement the authors aligned the UCD process with the product development lifecycle. In the course of time it showed evidence that there was no need for major changes to the product development lifecycle anyway (in contrast to the initial reservations of management). In practice the idea of the Meta Workflow and the Meta Tasks

aligned well with the idea of software components, even though they don’t necessarily match.

The Meta Workflow, separated into several Meta Tasks, represent the flow of work from the customers’ perspective. These Meta Tasks and their implicit goals are defined in a user-focused way with users’ intentions and expected system responses in user goal use cases. The characteristics of the personas, as well as the description of their intentions and expected system reactions (in contrast to concrete system interactions) imply requirements that directly feed the existing requirement engineering process.

As mentioned above each use scenario represents a specific part of the user’s daily sequence of work. To connect the user workflows with the requirement engineering process, each use scenario was captured in a single requirement, called “Use Case Requirement”. Each use case requirement is later on decomposed into development requirements by the requirements engineer, the architect, the PAs and developers cooperatively. Thus, there exists a relationship of one use case requirement to multiple development requirements. The sum of development requirements that are mapped to the use case requirements represents the elements of a user’s workflow from a development point of view.

The concept team uses the use scenarios as a basis to create user interface designs. Therefore, the UI Solution Papers are linked to the use case requirements they address. Furthermore, each concept is captured in multiple development requirements and fed into the requirements database. Hence, there is also a relationship between use case requirements and development requirements, representing the solution from a design perspective. Figure 3 shows the relationship of documents and requirements.

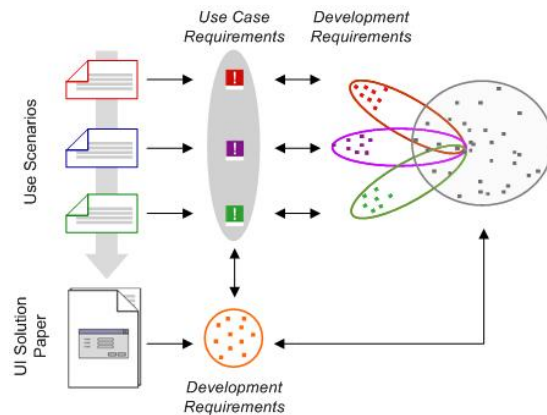


Figure 3. Interplay of Use Scenarios, UI Solution Paper and Requirements

In addition to the use cases, the developers are provided with the UI solution paper along with the list of non-functional requirements related to the use cases. The requirements connect the results of the analysis and conceptualization phase with the

requirement engineering part of the development process. The development requirements get a contextual coherence by being clustered through the use case requirements. An advantage is that they are not detached from each other.

The use scenarios and the corresponding use case requirements support the decision making process and help to prioritize and to select requirements for the product release specification:

- Based on use scenarios it can be decided which workflows must be implemented to support the targeted user group. The corresponding use case requirements point to the corresponding development requirements.
- Vice versa, it is now obvious if specific development requirements influence a use case requirement and whether it influences the corresponding use scenario, which could have an immanent impact on the user's integrated workflow.

Thus, the result is an integrated process that combines best practices of requirements engineering with the activities of the UCD process likewise. The users' workflows are now traceable like "regular" requirements.

The strongpoint of this approach is that software engineers and project managers now got results from the UCD in a requirement format that they were used to and it also allowed keeping the existing process models and CASE-Tools.

2.4. Validation

The validation phase is an integral part of the UCD process that exists in parallel to the other phases. It offers a "quality check" of requirement acquired so far. The activities of the validation phase can run in parallel to other UCD activities, as long as it is guaranteed that the follow-up activities start after their required documents have been validated.

Within this project, the authors introduced two different types of validations: "In-process" and "On-site" validations. In-process validations can be performed with less detailed documents to check fundamental assumptions. They are performed by internal stakeholders without the explicit involvement of external participants, e.g. users or customers, whereas on-site validations need a certain level of detail to create notable outcomes. On-site validations always include the target audience of the solution such as the users or the customers of the system.

Especially in process validations turned out to be very effective and time saving. For example, the involvement of Product Analysts within the UCD process provided an opportunity to validate requirements without the necessity to involve users or customers. In most cases the domain knowledge of the PAs was sufficient to provide valid and valuable feedback to the quality of the generated requirements

and to make educated decisions. However, not only the PAs performed and participated in validations but also the other document stakeholders validate their input material similarly. For example, first PAs performed the task analysis and created the use cases from the analysis results. Then these use cases are being validated with the users to ensure that all user requirements have been considered and that the use scenarios match their workflow. As another example, the concept team creates design concepts based on the use cases. Then, the designs are being validated by the PAs by focusing on the user's requirements and on the technical feasibility by the development team.

Of course, the focal point during the validation is always the users and their workflows since it is the integral part of in a UCD approach. It has to be validated if all user requirements have been considered and if all made assumption regarding the users' behaviors and needs prove to be true. Typical questions were: Is the definition of the target audience correct? Are they described correctly? Are there any gaps? Do the made assumptions meet the users' needs? Does the solution fit the users' expectations? Etc. Therefore, several validation activities have been accomplished by different stakeholders as shown in Table 1.

Table 1. In-Process and On-Site Validations

<i>Output document</i>	<i>Created by</i>	<i>Validation (in-process) performed by</i>	<i>Validation (on-site) performed by</i>
Meta Workflow	Marketing	Product Analysts	Customer
Use Cases (any types)	Product Analysts	Customers	User
Concepts & Conceptual Prototypes	Concept Team	Product Analysts & Development	User & Customer
Designs	Concept Team	(Product Analysts)	User
Functional Prototypes & Pre-Releases	Development	Product Analysts, Concept Team & Test Team	User & Customer
Release Version(s) of the Solution	Development	Product Analysts, Concept Team & Test	User

The different types of use cases are the central analysis documents. They are being used for all subsequent validations activities. They are also the basis to generate test cases to check if the solution fulfills the users' requirements.

The strict appliance of validation activities concerning the (up to now regarded as) informal user oriented parts of the analysis in the UCD process model ensure the quality of results in (especially regarding the usability of the product) and simultaneously decreased the adjustment and maintenance efforts likewise.

3. Achievements & Findings

The UCD Approach as applied within this project highlights the advantages not just in the sense of better products but also for the organization itself.

Looking back it showed that the two main goals of this project have been achieved successfully. User feedback gave first evidence that the new version of the software solution fits the users' needs and supports their workflow more efficiently. Additionally the project management saw that an UCD process can create substantial advantages for an organization. As the reasons for success, the following highlights could be named:

- Use Cases create a common understanding about the market, the customers and the users between all parties involved.
- Use Scenarios and Use Case Requirements connect the UCD Process with the Requirement Engineering Process which ensures the traceability.
- The traceability of Use Case Requirements ensures the development users' workflows and the consideration of users' requirements.
- Concepts and designs are created based on use cases and therewith represent solutions that fit users' requirements and workflows.
- Based on the modular analysis results, the organization was able to derive a dynamic product portfolio which can be handled more easily and could be adapted to specific market and customer needs.
- The Marketing gets documents which speak the language of the customers as a foundation for discussion.
- The support of Use cases enables the requirement analysts to capture and validate customers' expectations and can be used as communication medium throughout the product development lifecycle.
- The combination of an interdisciplinary process team brings together a wide knowledge and understanding of the product development lifecycle.

Today, the UCD process methodology takes place in the organizations' product development process and will influence the next versions of products.

4. Conclusion & Outlook

The User Centered Design (UCD) approach and its activities are important and valuable not only during the design and implementation phase. Bringing users' workflow into the project focus, structuring documents, and organizing design and development processes accordingly, the software development company was able to gain valuable improvements. If UCD is regarded as an entire proceeding, and its underlying concepts and principles are established, then not only one product provides increased ease-of-

use and usability, but the whole structure of the product portfolio does.

The proceedings and results of this study show advantages of an integrated UCD approach from a software producer's point of view. Even if this study could not cover all aspects of implementing a UCD process into an organization it highlights the notable improvements and links between UCD activities and documents to the product development lifecycle.

An analysis of organizational conditions and the correlative impacts, as well as the evaluation of quantitative data, have to be considered for future work. The authors believe there is still a lot of space for development and further research.

Next, the authors will examine other areas this methodology can be established in. Until now the described method concentrates on how to define and validate usability and software engineering relevant data within the process. The next improvement will enhance the domain of quality management, which is already involved implicitly but not really put into focus. This will include the unification of requirements from usability and software engineering with less formal but nevertheless important constraints from design and marketing aspects. Domain experts are already incorporated as stakeholders but it is still challenging to find new concepts for design, usability, interaction, technical platforms and domain expertise at the same time.

10. References

- [1] R.G. Bias, D.J. Mayhew, *Cost-Justifying Usability. An Update for the Internet Age*, Morgan Kaufmann, San Francisco, CA, 2005
- [2] Beyer, H. & Holzblatt, K. *Contextual Design*. Academic Press, San Diego, CA, 1998
- [3] Cooper A. and Reimann, R. *About Face 2.0, The Essentials of Interaction Design*. Wiley Publishing Inc., Indianapolis, IN, 2003
- [4] Cockburn, A. *Writing Effective Use Cases*. Addison-Wesley, London, UK, 2001
- [5] Gulliksen, J., Blomkvist S., Boivie I., Cajander Å, Göransson, B., Persson J. *Key Principles for User-Centered System Design*. In Proceedings for INTERACT 2003, Zürich, Switzerland, 2003
- [6] Karat, J. & Karat C. M. The evolution of user-centered focus in the human-computer interaction field. *IBM Systems Journal*, Vol. 42, No. 4 2003, 2003
- [7] Mayhew, D.J. *The Usability Engineering Lifecycle*. Morgan Kaufmann, San Francisco, CA, 1999
- [8] Nielsen, J. *Usability Engineering*. Academic Press, Chestnut Hill, Mass, 1993