The Usability Engineering Repository (UsER)

Marc Paul, Amelie Roenspieß, Tilo Mentler, Michael Herczeg

Institut für Multimediale und Interaktive Systeme (IMIS)
Universität zu Lübeck
Ratzeburger Allee 160
23562 Lübeck
paul@imis.uni-luebeck.de
roenspiess@imis.uni-luebeck.de
mentler@imis.uni-luebeck.de
herczeg@imis.uni-luebeck.de

Abstract: The Usability Engineering Repository (UsER) is a tool system designed and implemented in prototypical state to support the analysis, design and evaluation of interactive systems. For this purpose, UsER provides method modules covering different aspects of analysis, design, implementation and evaluation. While created in a linear, document-like structure, contents can be interlinked in order to point out semantic interrelationships, ease navigation and allow tracing requirements forward and backward in the process. The modules can be configured and used as needed and none of them is mandatory. We will introduce UsER using the Human-Centered Design (HCD) Process as example.

1 Introduction

Various development processes for software engineering exist. However, the user-centered analysis, design and evaluation of interactive systems are deficiently taken into account in many projects. Either the process model does not support considering users, tasks and overall context or integrated tooling is missing. We combined several approaches in the areas of Software Systems Engineering, Requirements Engineering and especially Usability Engineering to support the whole process of development. The current result is an analysis and modelling environment as development repository called UsER (Usability Engineering Repository). It allows to flexibly combine different tools and methods depending on project-specific needs as well as to trace and reuse all information raised during the development process. UsER supports the process of human-centered design (ISO 9241:210) according to the six key principles:

1. the design is based upon an explicit understanding and documentation of users, tasks and context;
2. stakeholders can be involved throughout design and development;
3. the design is driven and refined by user-centered formative evaluation;
4. the process is iterative;

5. the design addresses the whole user experience;
6. the design team provides multidisciplinary skills and perspectives.

Most software developing companies meanwhile recognize how important usable software is for their customers, but not how to reach this goal. Usability is not simply an additional requirement to be implemented at the end of the development. Usability Engineering provides a wide range of methods and systematic approaches for the support of development, for example the Goal-Directed-Design [CRC07], the Usability Engineering Lifecycle [Ma99] or the Human-Centered Design (HCD) Process [ISO11], to name but a few. All of these need to be applied throughout the whole development process. We decided to design and develop UsER as a modular tool environment so it can be adapted flexibly to any existing process.

2 UsER Modules in a Sample Process

To provide a systematic and predictable approach to software development, we support every step of the HCD process (as an example process) with certain modules (see Fig. 1). In order to benefit from this process, all of its results may be transferred to the implementation process. Requirements are particularly well suited for this task.

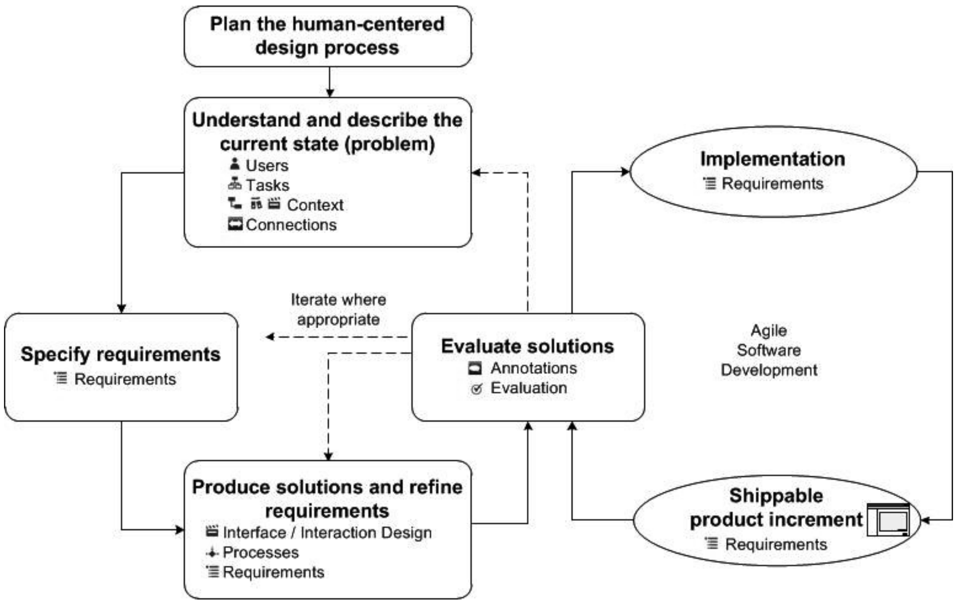
Plan the human-centered design process
Understand and describe the current state (problem)
Users
Tasks
Context
Connections
Implementation
Requirements
Iterate where appropriate
Specify requirements
Requirements
Evaluate solutions
Annotations
Evaluation
Agile Software Development
Produce solutions and refine requirements
Interface / Interaction Design
Processes
Requirements
Shippable product increment
Requirements

Figure 1: Extended Human-Centered Design Process

A UsER module supports certain aspects of a usability engineering method. Information gathered in one module can be interlinked with information in other modules. These references help to understand development decisions made during analysis, design or evaluation, as they can be retraced directly from the requirements concerned.

A development project in UsER is organized as a linear structure like a typical development document. Each chapter or section provides the functionality of a specific module (see Figure 2). These can be arranged via drag&drop to model contract and development documents.The resulting structure itself can be stored and later reused as a template for future projects. The modules realized so far support analysis and design as well as requirements management throughout the whole software development process.

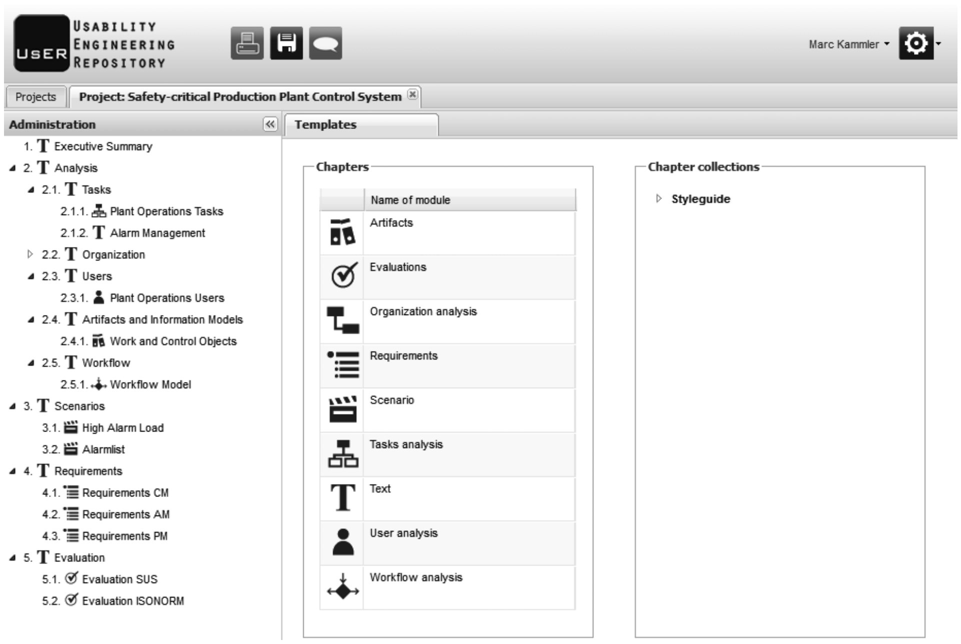

Figure 2: Selected combination of modules in UsER

2.1 Analysis of the Domain and the Users

Hereafter, we will have a look at the functionality of some modules following the approach of HCD.

User Analysis

In order to understand the context in which application software is supposed to be used, it has to be analyzed together with the stakeholders. The first idea for this module was to create and manage personas [CRC07, PA06], which could then be reused in the other modules. During development we realized that it would be helpful and practical classifying user descriptions on different levels of detail [PA06]. User classes or less abstract stereotypes can now be created as organizational roles or target groups. The information collected in user classes can then be reused by inheritance and refined for more detailed user descriptions. Thus, user descriptions can be defined on an abstraction level according to the needs of the project [PA06]. Additionally, the module has a progress bar indicating roughly how much important information has already been specified in the user description. Including field data can be important for user descriptions. UsER supports this by providing so called "behavioral variables" [CRC07]. These are user attitudes and aptitudes which are relevant for the project, e.g. whether the users prefer simple interfaces over customizable ones. The answers are mapped onto a bidirectional scale, visualizing trends and clusters which can be used to describe characteristics of fictional users based on field research data. The data might even be gathered semi-automated from an evaluation module. Specified for single or several projects, user descriptions also contain user goals, which can become requirements in order to be taken into account for the later specification. It is indispensable to determine and analyze the tasks future users are supposed to carry out with the system [Co99, He09]. The module for user analysis already allows task description on an abstract level, also called "external tasks" [He09] in operational contexts. These can be described in more detail in the module for task analysis, while the artifacts used can be specified in the module for artifact models.

Task Analysis

Tasks may be decomposed into subtasks using the model of Hierarchical Task Analysis (HTA) [AD67]. The integrated list editor allows direct graphical manipulation of the tree representation as well as keyboard-only input. HTA is an effective tool for visualizing and understanding tasks [Be10] as it shows tasks in the context of their higher purpose. A task description contains some basic information like pre- and postconditions, duration, criticality etc. As mentioned before, artifacts (work objects) required for a task can be described in a module of their own. Additionally references to organizational roles may be established, which allow the description of team-based tasks and lead to the method called HTA-T (HTA for Teams). A UsER-wide component allows interlinking any entities of all modules.

Artifact Models

From the user's perspective, it often is not the application which is relevant but the work objects to be manipulated [PA06] like documents or physical work objects. UsER supports the modeling of artifacts with names, an optional description and arbitrarily configurable attributes.These can serve as a basis for information models or help to describe detailed context information by linking to other modules, e.g. for defining who in an organization works with which artifacts or for which task certain artifacts are required.

Organization Analysis

Information about operational structures is often depicted in form of organization charts. The handling of this module is similar to existing applications like Microsoft Visio. Organization units, roles and staff positions can be described in this module. Every piece of information already collected – e.g. users or tasks – can be assigned to the elements of the organization chart to get a good overview and deeper understanding.

Scenarios

Alternatively or additionally it is possible to describe the application in a more informal textual form. Inspired by the approach of Scenario-based Design by Rosson and Carroll [RC01], this module supports the description of problem scenarios. Furthermore, it allows to enrich every scenario by adding pre- and postconditions. All information collected so far (users, tasks, etc.) can be linked to these descriptions. This module is also suitable for the transition to the next step of the HCD process. As a rich text application the module allows to include graphics and other media.

2.2 Specifying Requirements

For a comfortable transition to the next step of the HCD process there is an interface integrated in the scenario module for deriving requirements from each problem scenario. Here it is important to articulate these requirements in an abstract way like recommended by Goal-oriented Requirements Engineering [La00, La01, Po08]. All requirements gathered from the scenario module can be processed inside this module to create formal requirements. It will be possible to import or export requirements from or into other requirements engineering environments through the standardized ReqIF-format.

The requirements module collects all requirements within one project in a sortable table. It allows decomposing a requirement into sub-requirements. Furthermore, it is possible to add more information to the requirements, like source, reason, type (bug, new feature etc.) or a category to sort the requirements by system components.

2.3 Create Solutions and Refine Requirements

The idea in this step of the HCD process is to refine the abstract requirements (goals) previously collected by describing the activities in a textual way. These textual descriptions (“design scenarios” [RC01]) work well for discussing solutions with the stakeholders. They are used to specify how a goal can be met [Ro98]. Because of the different ways imaginable for achieving a goal, it should be possible to describe whole use cases with a main case and alternative cases. The module we are using for this step is the scenario module already used for describing the problem situation at the beginning of the process. To attach several scenarios to one use case it is possible to include them in different chapters within a project. Scenarios can be interlinked using the hypertext function. For refining the scenarios continuously we integrated the external mockup tool Balsamiq® which supports the fast and flexible creation of low fidelity prototypes.

2.4 Evaluate the Solutions

Two of the key principles of HCD are involving the user and evaluating the developed solutions together with them. To support this as well as an iterative refinement process we integrated an evalution module which allows formative and summative evaluation of the software using standardized or specialized questionnaires together with an annotation function which allows comments for every element of content. By this, solutions can be collaboratively refined and iterated. As the results of questionnaires are shown in UsER and the annotations are sorted chronological, decision processes can be retraced. When a solution finally meets the user requirements, the concepts can be transferred to the developers for implementation.

3 Summary and Future Work

The Usability Engineering Repository (UsER) is an integrated modular framework to support various user centered development methods and processes for interactive systems in a flexible way. Modules for user, task and organizational analysis can be combined with modules for scenarios, interaction design or evaluation. As an open architecture UsER can be extended by additional modules. For instance, we are currently working on tool support for the analysis, design and evaluation of safety-critical systems.

References

[AD67] Annett, J. and Duncan, K. Task analysis and Training Design. Occupational Psychology. 41, July, 1967, pp. 211–221.

[Be10] Benyon, D. Designing Interactive Systems: A Comprehensive Guide to HCI and Interaction Design. Harlow, England: Pearson Education, 2010.

[Co99] Constantine, L. Software for use: A practical Guide to the Models and Methods of usage-centered Design, 1999.

[CRC07] Cooper, A., Reimann, R. and Cronin, D. About Face 3: The Essentials of Interaction Design, 2007.

[He09] Herczeg, M. Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchs-taugliche interaktive Computersysteme. München: Oldenbourg Wissenschaftsverlag, 2009.

[ISO11] International Organization for Standardization. ISO 9241-210:2010 Human-centred Design for Interactive Systems, 2011.

[La01] Lamsweerde, A. Goal-Oriented Requirements Engineering : A Guided Tour. Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, 2001.

[La00] Lamsweerde, A. Handling Obstacles in goal-oriented Requirements Engineering. IEEE Transactions on Software Engineering, 10, 2000, pp. 978–1005.

[Ma99] Mayhew, D.J. The Usability Engineering Lifecycle : A Practitioner's Handbook for User Interface Design. San Francisco, CA: Morgan Kaufmann, 1999.

[Po08] Pohl, K. Requirements Engineering. dpunkt.verlag, 2008.

[PA06] Pruitt, J. and Adlin, T. The Persona Lifecycle: Keeping People in Mind throughout Product Design. San Francisco, CA: Morgan Kaufmann, 2006.

[Ro98] Rolland, C. Guiding Goal Modeling using Scenarios. IEEE Transactions on Software Engineering 12, 1998, pp. 1055–1071.

[RC01] Rosson, M. and Carroll, J.M. Usability Engineering: Scenario-Based Development of Human-Computer Interaction. San Francisco, CA: Morgan Kaufmann, 2001.