

# Tool-Supported Usability Engineering for Continuous User Analysis

Anna Hüttig<sup>(✉)</sup> and Michael Herczeg

Institute for Multimedia and Interactive Systems, University of Luebeck, Luebeck, Germany  
{huettig,herczeg}@imis.uni-luebeck.de

**Abstract.** There is a large set of methods and measures to strengthen the usability of interactive products; however the practical application of usability engineering still seems to be difficult and is not practiced widely. As main problems we identified a low level of guidance, insufficient flexibility of methods and lack of awareness of results within the project team. We will discuss these problems and present a new approach of performing usability engineering in practice as collection of easy to handle tools that can be combined and integrated throughout the whole development process. The contribution places particular attention to the area of user analysis and how to support this important aspect of software development within our system Usability Engineering Repository (UsER).

**Keywords:** Persona · Software engineering · Tooling · Usability engineering · User-centered design · User analysis · User class

## 1 Challenges of Performing Usability Engineering

In theory, there are plenty of methods, processes and measures to analyze and strengthen the usability of interactive products [1, 3, 13, 16, 19, 26]. However, the practical application of usability engineering still seems to be difficult and is therefore often neglected or practiced insufficiently [8, 17, 23, 25, 27, 29]. We identified three main problems that can be seen as obstacles for the practice of usability engineering within authentic project contexts: no or low level of guidance, insufficient flexibility of methods and lack of awareness of results within the project team (Table 1).

### 1.1 Guidance in Method Application

The first point, the low level of guidance, refers to the question, how software analysts, designers, programmers and testers can be enabled or supported to apply methods of user analysis. This often starts with a lack of understanding of the methods, the difficulty of finding a reasonable starting point or the challenge of a meaningful adaption of methods chosen for the particular project or domain [17].

The efforts of method application depend on the specific methods used. Within certain application domains or under certain project management conditions, some methods are better suited though being more labor-intensive and expensive. Methods with low thresholds or efforts often provide weaker results. Another important factor is the experience and

routine of the persons performing the usability engineering processes. Human-centered design skills are not trained in most software development enterprises [25, 29].

**Table 1.** Main challenges of practical application of usability engineering (UE) methods

Challenge	Implied obstacles for the user who may not know how to	Frame conditions
The user of UE methods receives little <i>guidance</i> for the application	<ul style="list-style-type: none"> <li>... understand the method itself</li> <li>... find a suitable starting point</li> <li>... adapt the methods for the needs</li> </ul>	<ul style="list-style-type: none"> <li>• (special) requirements of UE method</li> <li>• experiences of executing persons/project team</li> <li>• existing and habitual software engineering processes</li> <li>• budget</li> <li>• available time</li> </ul>
The UE methods do not seem to be flexibly <i>integrable</i>	<ul style="list-style-type: none"> <li>... integrate the measures into existing processes</li> <li>... combine the methods to foster the linkage of developed results</li> </ul>	
Developed results and models lose their potential since they are not <i>perceived</i> by the team	<ul style="list-style-type: none"> <li>... communicate the results</li> <li>... achieve acceptance for the results</li> <li>... foster the utilizing of results</li> </ul>	

## 1.2 Flexibility in Method Application

The second problem addresses insufficient flexibility and the question how usability methods can be integrated into existing processes. Companies often are not able or willing to change their development processes. Others seem to be quite helpless, how to change existing development workflows for usability methods or how to see interfaces between usability engineering activities and standard software engineering procedures [17].

Fading results are a related problem, meaning that the potentially expensive application of usability engineering methods becomes worthless in later phases of the process because of deficient integration of the results in ongoing or following development phases. This is related to the issue that the results of the analyses need to be merged and cross-referenced to provide a coherent overview of the project and problem scope. A set of single, self-contained methods without bringing the results together cannot reveal all relevant aspects.

## 1.3 Awareness for Developed Results

The third problem area, the lack of awareness, stresses the point, that especially small or medium sized enterprises do not have the resources to hire their own usability experts [29]. But even if they have a team or person responsible for this, the efforts become useless if the results are not communicated to all project members or if they are not accepted by them. Especially user analysis and user modeling become rarely interweaved with standard system development processes. In practice, even when flexible

and low-threshold methods like personas have been applied, the outcome is ignored, neglected or simply gets lost during the further process [17].

The classical split of roles within software project teams that can be found in conventional as well as in more agile software engineering approaches implies certain communication paths and processes. These require the frequent exchange of information between domain experts or team members with various backgrounds. A typical project team could consist of a consultant who is in contact with relevant stakeholders and end users of the software product that has to be developed. The consultant discusses his or her insights with product designers who incorporate the findings in the conceptual design of the application. These concepts have to be implemented later by a team of software developers. Along this communication path through groups with different perspectives on the problem domain, the typically quite unstructured, but authentic statements of end users have to be transformed into technical implementation specifications. This process demands the need for a common and shared understanding of the project's scope. The exchange and alignment of information has to be supported often among teams that do not consist of members that are particularly educated in mediating usability ideas.

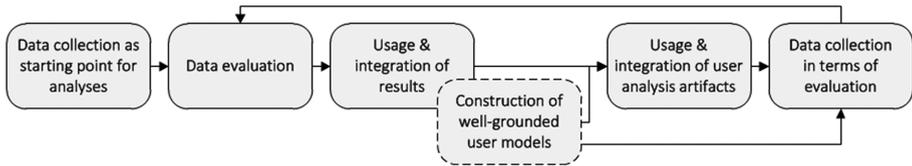
#### 1.4 Novel Approach to Usability Engineering in Practice

The above mentioned challenges can be approached by making usability engineering methods more practical to use and more capable to be integrated and combined. They shall not appear as particular procedures that have to be performed at some point in the process, but as manageable activities that can be integrated flexibly and that can be used and extended at any time throughout the development process. To achieve this a strong *tool support* for usability engineering methods is needed. Well-designed tools can decrease the threshold of method application and give sufficient guidance in method execution. Therefore, they have to be easy to use and flexible enough to be integrated into existing project settings. Additionally, they can help to create a holistic view of the application domain and the target group of end users if the results are combinable. By this, a more understandable and more appropriate model of the application project can be spread out over all software development processes in order to support the common understanding among all project members. We found this tool approach to be a promising way to foster *manageability* and *extensibility* of methods and to strengthen *embedding* of usability results in major contexts.

## 2 Continuous User Analysis

Concerning the analysis of users, specialized tools and their combination with instruments for other usability engineering methods can support all phases of user analysis and foster a *Continuous User Analysis* throughout the project and its phases with steady user focus among the project team. The term *continuous* shall emphasize that user analysis should not be seen as a closed method within some temporary phase. In the sense of user-centered design processes, the user has to stay in focus throughout the whole development and all design decisions should be based on or at least be balanced against user needs.

Figure 1 shows a possible structure of the different steps of user analysis measures in terms of usability engineering. The first step (“Data collection as starting point for analyses”) is the main foundation for further analyses. This covers the huge set of field study techniques and methods from simple third party research over user interviews to complex supervision studies [6, 15, 18]. Collected data has to be reviewed, evaluated and prepared for further usage and integration in follow-up processes (“Data evaluation”).



**Fig. 1.** Possible division of phases of user analysis for usability engineering. The usage of user models is one suitable method within this process of *Continuous User Analysis*.

User data can be abstracted into user models which are a basic concept of user analysis and the realization of steady focus on user needs in all project phases. User models are well-suited to be utilized and applied in all following conception and development phases in many ways. Serving as independent artifacts, they can be used for discussions and reviews and be combined with other usability methods. Problems that have to be faced – besides the quite common complete lack of user models in general – are the creation of unsuitable and ill-defined models and the insufficient transfer of user models into the ongoing system development process.

Concerning the modeling of users, several techniques have been developed: While in usage-centered design more abstract models like actors or role descriptions have been used to analyze usage patterns [2], user-centered design aims for more realistic models. An obvious first step to handle the diversity of users is their division into user classes [30]. Main distinctive features can be – depending on the product’s context – the user’s goals concerning the target system, the technical or use-oriented level of experience or the organizational role of the users [10, 22]. Latter shows the tight interplay between role descriptions and user classes. Class descriptions can reach from simple, often assumption-based category depictions to detailed, well-grounded user profiles [9]. However, user classes remain abstract and are often not telling enough about the real user needs. More vivid are user descriptions that depict a single concrete, fictive person as a representative for a specific group of users. Popular methods are the modeling of different types of personas [3]. They consist of detailed, mainly narrative descriptions of fictive persons often based on extensive studies and data of real people [22]. The potential and handling of personas has been described extensively in literature [20].

Another form of modeling is the archetype (e.g. “the blind user of a vending machine”) or – with rather negative connotation – the stereotype (e.g. “the helpless elderly user of a vending machine”). Here, the archetype shall be understood as a denomination for a putative precisely defined class of persons with specific characteristics that is – at least along general lines – interindividually valid. The delimitation to user classes and to concrete user descriptions is smooth and often unclear. Archetypes represent, like user classes, a type of

users, though archetypes are more precisely and demonstratively defined. They are suited to mediate specific characters of a category, but also tend to caricatures so that the values of these descriptions are often quite limited [10]. However, an advantage of archetypes is their simple creation since only few or even no user data has to be collected [7, 28]. Furthermore, so-called Extreme Characters are described as narrative, extreme personalities with exaggerated emotional attitudes [5].

User models only unfold their entire potential when being integrated in the ongoing development process during and after their creation and refinement. Especially personas seem to be helpful. Process models like the Goal-Directed Design [3, 4] or the Concept Development Process for requirements engineering [27] regard user descriptions within the development process and pursue the idea of not letting the models become fixed and more or less dead artifacts after their creation. Instead, they should permanently be present, accompany and support the development process and teams together with other usability methods and also be evolved and enhanced during the design and implementation phases of product development.

The necessity for continuous consideration of user data is obvious since the software product is intended to address the observed user needs – and will likely fail if not fulfilled. So, steady extensions of user focused analyses – not necessarily in form of user models – and combinations with analysis methods concerning other usability engineering artifacts as tasks and requirements are obviously meaningful. Especially, stronger interweaving of requirements and user needs seems promising. The systematic, transparent and comprehensible coupling of usability engineering and requirements engineering is currently not sufficiently practiced (“Usage & integration of user analysis artifacts”). For reviews of first prototypes or in context of formative evaluations of the software product (“Data collection in terms of evaluation”), the created user models can be consulted, too. Several techniques are named in literature [22].

### 3 The Usability Engineering Repository (UsER)

We have implemented the ideas described above within our Usability Engineering Repository (UsER). UsER is a web-based system supporting classic as well as agile software development processes by providing modules representing usability engineering methods (Fig. 2). All information gathered is modeled in form of entities by one

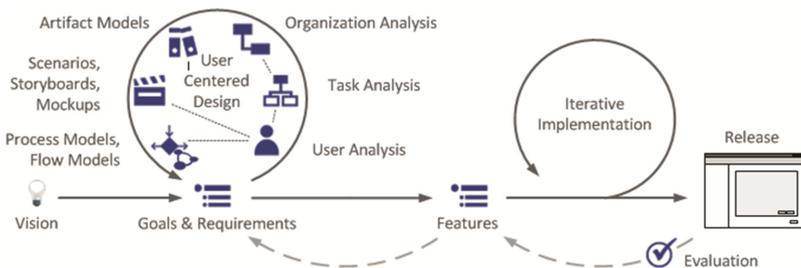
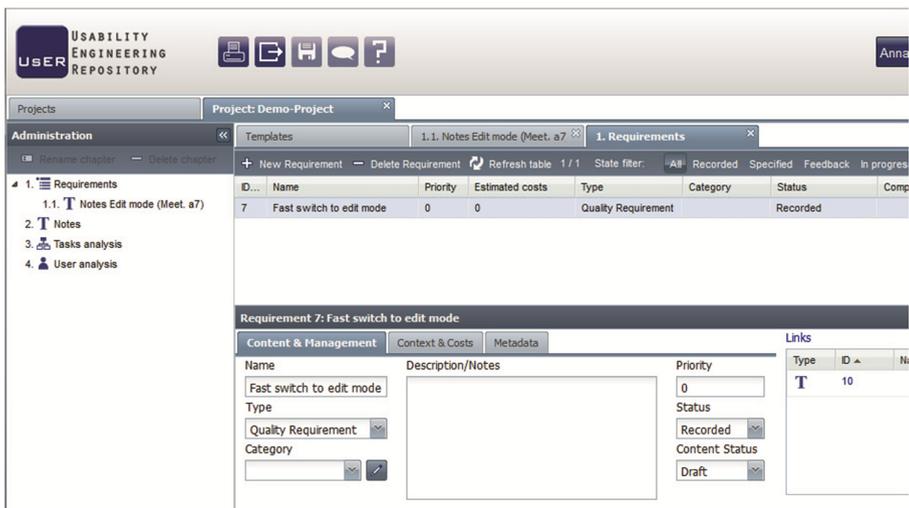


Fig. 2. Possible usability engineering process supported by UsER modules

of the UsER modules and may be interlinked with entities from the same as well as other modules. Besides the versatile cross-references between entities, the development teams may select and combine the modules as needed [11, 15, 21].

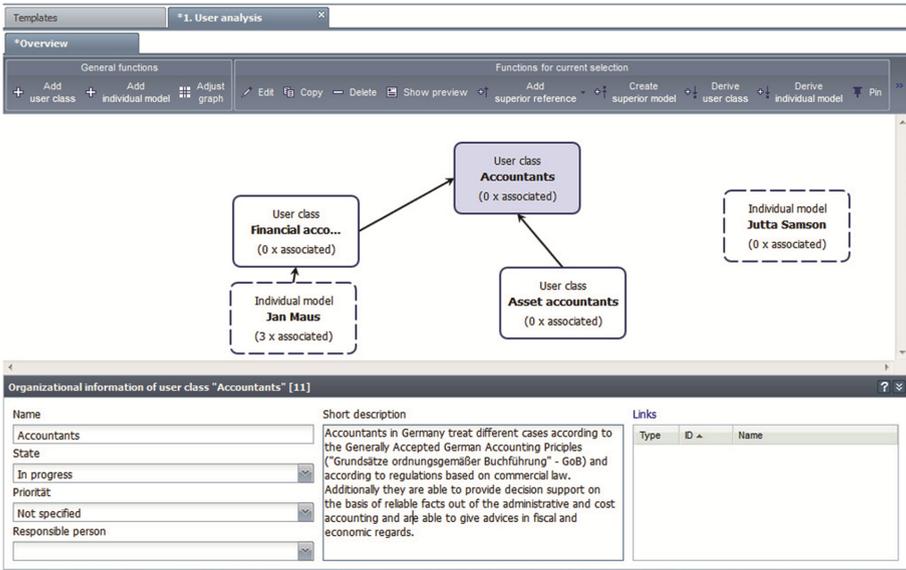
Usually a project will be organized in a classical linear structure as in conventional development documents. Standard development processes therefore will be supported more easily. Every “chapter” in “a UsER document” provides the functionality of a specific method module. They can be dragged into the linear organized administration area of the current development document (Fig. 3). By this, hierarchically structured specification documents can be constructed and the development document itself becomes a starting and anchor point for the different development tools. This is an important change of paradigm, leading from classical text- or table-centered specifications into active and semantically modeled living documents with a classical linear outer structure and an inner structure of semantic networks for holistic analysis, design and evaluation of interactive systems. Apart from the user analysis module that will be described more detailed, UsER offers a variety of other modules, e.g. for collecting and managing requirements, analyzing organizational structures or describing scenarios. Modules can be used in several phases within an engineering process.



**Fig. 3.** This screenshot of UsER shows the opened Requirements Module next to the document management panel (on the left). The detail area of each module includes the linking widget (on the bottom right) offering the functionality to define links between UsER entities.

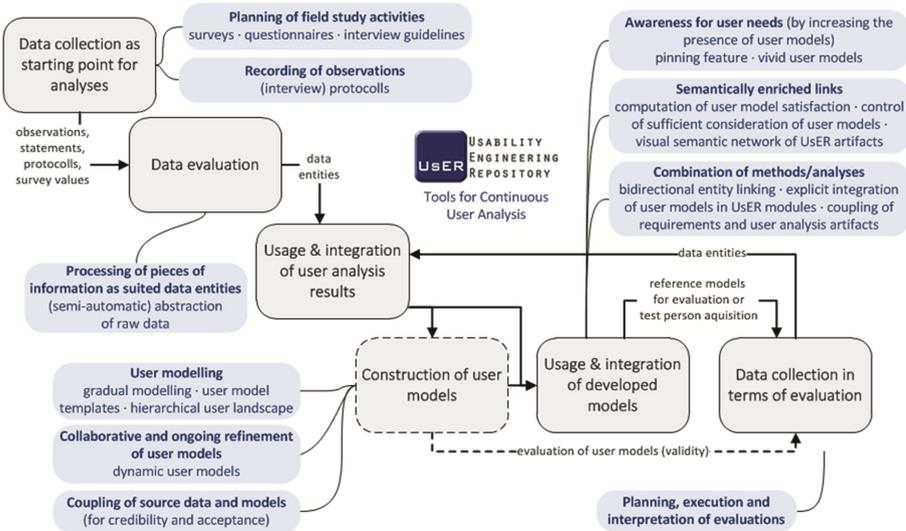
## 4 Integration of Continuous User Analysis Within UsER

As discussed, user analysis is a central aspect of user-centered design processes and user models act as helpful instruments to support all phases of software development. Therefore, tool support for user analysis and user modeling should be a substantial component within frameworks for the development of interactive systems. UsER in particular –



**Fig. 4.** The user analysis module in UsER. This screenshot shows the graphical user landscape editor. Additional views exist for the content-related design of single user models and for the definition of user model templates.

with its collection of several usability engineering instruments – offers a suited environment to implement the idea of *Continuous User Analysis*.



**Fig. 5.** Tools for *Continuous User Analysis* within UsER as a platform for usability engineering instruments. The tools shall support all phases of user analysis as named above.

The development of the User Analysis Module (Fig. 4) in UsER [12, 24] was the first step for a systematic integration of user analysis artifacts within the system. At the beginning, the focus lay mainly on building user models, so that after several iterations the instrument supported software designers in creating and improving user models by providing a systematic approach of gradual two-stage user modeling. This allows constructing e.g. abstract user classes or more detailed stereotypes and quite individual descriptions like concrete personas – means user models on different levels of abstraction.

Utilizing the potential of UsER as a platform and toolbox for usability engineering methods, we then fostered the integration of user models into other modules next to the user analysis tool and into the UsER core system (Fig. 5). We now enable the explicit integration of modeled users as performers of tasks in task analyses, as vivid actors in scenario descriptions or we let the models be built in into flow models or organigrams as representatives for organizational entities.

As a central feature, UsER allows the definition of arbitrary bidirectional links between any two UsER entities. So, each user model can be associated with any defined usability engineering artifact, like for example requirements. Beyond, as one of the next steps, we will address a closer combination of requirements and user analysis artifacts. This shall include referencing user models as origin of a requirement and the introduction of user needs as new type of requirements.

Another aspect is the extension of the UsER linking concept itself: The bidirectional links (cross-references) between UsER entities shall be complemented with additional semantic annotations and attributes. These named links offer the potential for semi-automatic evaluations of analysis artifacts and their relationships. Within the scope of user analysis, one aim is the introduction of a kind of user (model) satisfaction: By semantically enriched links, we expect to be able to define validated rules and functions that process the status of entities linked to a user model and return cues for the satisfaction level of modeled users. One simple scenario could be described like this: A highly prioritized personas is linked to several requirements. Additional semantic annotations specify the links further – determining the linked requirements as must-have features for this certain persona. Now, if many of those requirements are marked as rejected (within the requirements module) or are not in advanced implementation stages, the project situation seems not beneficial for the persona's satisfaction. This can be reported within the system.

Furthermore, semantically enriched links offer more possibilities to control the sufficient consideration of relevant user models within all analyses and design activities.

In addition, we want to depict most links between UsER entities graphically. Especially after the introduction of semantic link attributes, a graphical overview on all UsER artifacts and their relationships among each other can serve as a suited entry point for the system. It enables a visual network of knowledge about the problem domain and fosters the clarification of the project scope.

To further foster the awareness for user needs and strengthen the focus on users throughout the entire project phase we are looking for methods to increase the presence of the models created within the system like integrating the models within different modules, as described above, and by integrating the models into the core system components of UsER. By a special feature, the Pin-Function (Fig. 6), selected user models can

be placed on the always visible project header within the UsER screen. An icon shows the most important facts about the modeled user and offers quick access to further information. This feature can be extended by letting the model become more alive by actively communicating to the current users of the system. For example, a persona could tell about its satisfaction level, i.e. how well its requirements have been addressed.

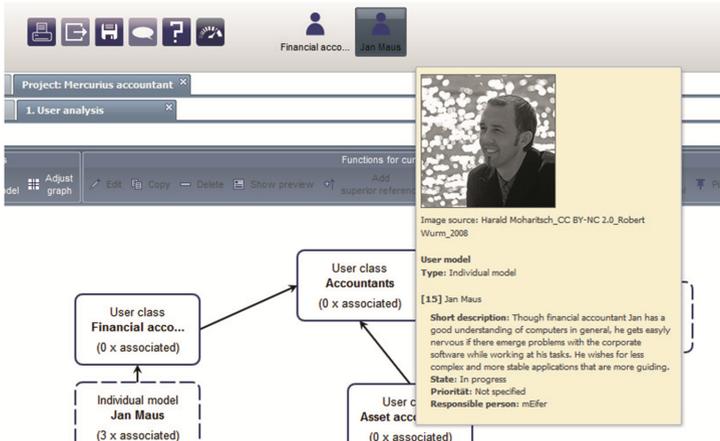


Fig. 6. The Pin-Function in UsER

User modeling and the propagation of the user models into the development process are just two steps within the process of a comprehensive user analysis. To get a well-founded basis for models, the overall process should start with data collection. To support this initial phase, we want to integrate modules for interview preparation and interview protocols as well as tools that support other field study activities such as contextual inquiries [1]. Furthermore, a tool for the construction of surveys has already been implemented and will be extended. Important collected pieces of information such as relevant user statements or evaluated survey items shall be integrated in the UsER system as entities that are connectable like all other UsER objects. By this, all collected facts can be referenced and used as credible pieces of justification for further analyses and user model design decisions.

To support the evaluation phase, tools for the planning, execution and interpretation of evaluations are modeled to be integrated as appropriate UsER instruments. User models can be involved within this phase as fictive reference users for the evaluations themselves or by consulting them for test person acquisition. Furthermore, UsER should offer the option to evaluate the user models themselves. Thereby, the compatibility of the modeled users towards the real user landscape can be validated.

## 5 Conclusions and Future Work

We discussed the demand for better tool support for usability engineering methods to overcome central problems of the weak usability engineering in practice. By introducing

flexible and easy to use instruments, we raise manageability of methods and the extensibility of analyses whose entities shall be easily embeddable in larger project contexts.

Especially, we consider user analysis activities as a central aspect of user-centered design processes that are quite often neglected. Our concept of Continuous User Analysis has already been established inside our Usability Engineering Repository UsER. It fosters steady focus on user needs and enables extensions of all findings. Further concepts and ideas are planned to be integrated to support all phases of user analysis. In the long term, this also includes expanding the systematic participation of real users within all project phases. They shall become active participants of the system UsER and shall be allowed to inspect and manipulate certain UsER artifacts by themselves. For that, more sophisticated user rights management and further collaboration features have already been planned. On the other hand, we like to take a closer look at the concept of Evidence-Based Usability Engineering as introduced by Metzker et al. [17] to support the sustainability of usability engineering experiences and knowledge.

UsER can be seen as a research platform for our concepts of *Tool-Supported Usability Engineering*. By integrating the idea of *Continuous User Analysis* into the system we like to contribute to more user-centered design and development processes that can be applied and controlled by teams within realistic project contexts. Positive evaluations of system components indicate that more representative evaluations of the overall system will verify our ideas.

## References

1. Beyer, H., Holtzblatt, K.: *Contextual Design: Defining Customer-Centered Systems* (Interactive Technologies). Morgan Kaufmann Publishers Inc, San Francisco (1998)
2. Constantine, L.: Users, roles, and personas. In: Pruitt, J., Adlin, T. (eds.) *The Persona Lifecycle*, pp. 499–519. Morgan Kaufmann/Elsevier, Amsterdam (2006)
3. Cooper, A.: *The Inmates are Running the Asylum*. SAMS, Indianapolis (1999)
4. Cooper, A., Reimann, R., Cronin, D.: *About Face 3: The Essentials of Interaction Design*, 3rd edn. Wiley, Indianapolis (2007)
5. Djajadiningrat, J.P., Gaver, W., Fres, J.W.: Interaction relabelling and extreme characters: methods for exploring aesthetic interactions. In: *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, pp. 66–71. ACM, New York (2000)
6. Goodman, E., Kuniavsky, M., Moed, A.: *Observing the User Experience: A Practitioner's Guide to User Research*. Elsevier, Waltham (2012)
7. Grudin, J.: Why Personas work: the psychological evidence. In: Pruitt, J., Adlin, T. (eds.) *The Persona Lifecycle*, pp. 643–663. Morgan Kaufmann/Elsevier, Amsterdam (2006)
8. Grudin, J., Poltrock, S.E.: User interface design in large corporations: coordination and communication across disciplines. In: Bice, K., Lewis, C. (eds.) *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1989)*, pp. 197–203. ACM, New York (1989)
9. Helander, M.G., Landauer, T.K., Prabhu, P.V.: *Handbook of Human-Computer Interaction*, 2nd edn. Elsevier Science Inc., New York (1997)
10. Herczeg, M.: *Software-Ergonomie. Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme*, 3rd edn. Oldenbourg, München (2009)

11. Herczeg, M., Kammler, M., Mentler, T., Roenspieß, A.: The usability engineering repository UsER for the development of task- and event-based human-machine-interfaces. In: Narayanan, S. (eds.) 12th IFAC, IFIP, IFORS, IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems. International Federation of Automatic Control, pp. 483–490. Las Vegas (2013)
12. Hüttig A., Herczeg M.: Tool-based gradual user modeling for usability engineering. In: Proceedings of the European Conference on Cognitive Ergonomics ECCE 2015 (2015)
13. ISO/DIN 9241-210: Ergonomics of human-system interaction - Part 210: human-centred design for interactive systems. Beuth, Berlin (2008)
14. Kammler, M., Roenspieß, A., Herczeg, M.: UsER: Ein modulares usability-engineering-repository. In: Mensch & Computer 2012: interaktiv informiert – allgegenwärtig und allumfassend, pp. 333–336. Oldenbourg, Munich (2012)
15. Lazar, J., Feng, J., Hochheiser, H.: Research Methods in Human-Computer Interaction. Wiley, Indianapolis (2010)
16. Mayhew, D.J.: The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. Morgan Kaufmann, San Francisco (1999)
17. Metzker, E., Reiterer, H.: Use and reuse of HCI knowledge in the software development lifecycle – existing approaches and what developers think. In: Hammond, J., Gross, T., Wesson, J. (eds.) Usability: Gaining a Competitive Edge/IFIP 17th World Computer Congress, Montréal, Québec, Canada, pp. 39–55 (2002)
18. Mulder, S., Yaar, Z.: The User is Always Right: A Practical Guide to Creating and Using Personas for the Web, 1st edn. New Riders Publishing, Thousand Oaks (2006)
19. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco (1993)
20. Nielsen, L.: Personas – User Focused Design. Springer, London (2013)
21. Paul, M., Roenspieß, A., Herczeg, M.: UsER – Ein prozessorientiertes Entwicklungssystem für Usability-Engineering. In: Boll, S., Maaß, S., Malaka, R. (eds.) Mensch & Computer 2013: Interaktive Vielfalt, pp. 181–190. Oldenbourg, Munich (2013)
22. Pruitt, J., Adlin, T.: The Persona Lifecycle: Keeping People in Mind throughout Product Design. Morgan Kaufmann, San Francisco (2006)
23. Richter, M., Flückiger, M.D.: Usability Engineering kompakt. Springer, Berlin (2013)
24. Roenspieß, A., Paul, M., Mentler, T., Herczeg, M.: Levels of abstraction for user modeling in the usability engineering repository UsER. In: Ahram, T., Karwowski, W., Marek, T. (eds.) Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics, AHFE, pp. 390–400. AHFE, Krakow (2014)
25. Seffah, A.: Learning the ropes: human-centered design skills and patterns for software engineers' education. *Interactions* **10**(5), 36–45 (2003). ACM, New York
26. Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human-Computer Interaction, 2nd edn. Wiley, New York (2011)
27. Sim, W., Brouse, P.: Empowering requirements engineering activities with Personas. *Procedia Comput. Sci.* **28**, 237–246 (2014)
28. Turner, P., Turner, S.: Is stereotyping inevitable when designing with personas? *Des. Stud.* **32**(1), 30–44 (2011)
29. Vukelja, L., Müller, L., Opwis, K.: Are engineers condemned to design? A survey on software engineering and UI design in Switzerland. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4663, pp. 555–568. Springer, Heidelberg (2007)
30. Wang, X., (n.d.): Personas in the User Interface Design, Alberta, Canada (2011). <http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/topic-wan-personas.pdf>