

Selecting Appropriate Evaluation Methods for Different User Centered Design Outcomes

Maximiliane Freymann
University of Paderborn,
C-LAB
maximiliane.frey mann@c-lab.de

Lennart Grötzbach
Siemens Business Services,
C-LAB
lennart.groetzbach@c-lab.de

Karsten Nebe
University of Paderborn,
C-LAB
karsten.nebe@c-lab.de

Motivation

In today's literature there are several established methodologies available that describe how to implement User Centered Design (UCD) in existing software development lifecycles [5]. Most approaches contain, that evaluation activities in UCD Processes are essential and that it is always important to check if the designed solution meets the users' requirements and to determine whether the solution is usable or not.

In Software Engineering (SE), evaluation activities are also fundamental, as they can be found in many of the existing models, like in the V-Model XT [6], the Spiral Model [1], Agile Development [2], etc. However, SE evaluation activities do not primarily focus on validating user-centered requirements, but rather focus on completeness and correctness of the implementation instead.

This means, user-centeredness cannot be ensured, until there is more focus laid on user related evaluation criteria. To achieve this, integration of SE and UCD is important at all levels or phases within the process, including the selection of appropriate evaluation methods.

Therefore, usability engineers are confronted with the task of providing appropriate methods (and tools for method selection) as well as user centered evaluation activities that can be applied and integrated appropriately in a software engineering process.

Approach

The authors provide an overview of user-centered evaluation methods that are available for different phases of a software development process, e.g. requirement specification and software evaluation and provide means to select the appropriate method for each phase.

The DIN EN ISO 13407 (Human-centered design processes for interactive systems) [4] defines four user-centered design activities: 'Understand and specify the context of use', 'Specify the user and organizational requirements', 'Produce design Solutions' and 'Evaluate designs against requirements'. The first three activities generate different types of user-centered requirements: Within the first activity, overarching usability requirements are defined, which are measurements or metrics that aim at the effectiveness, efficiency and satisfaction of the overall system. User Requirements are the result of the second activity. These requirements describe specific needs and information users need while performing a given task, e.g. details on workflow steps that have to be performed with the system. User interface requirements are generated as a result of the third activity and specify user interface details of the future system from a users' perspective, e.g. the 'Save'-button should be one click away and no option should be more than two clicks.

According to the definition by Zimmermann and Groetzbach [7], these three different results in terms of requirements are called 'Usability Requirements', 'User Requirements' and 'User Interface (UI) Requirements'. They can be evaluated using different UCD evaluation methods.

Depending on the results or requirements types, different evaluation methods are suited best to verify the results. In addition, the choice can be based on the criteria an evaluation method aims at. Usability can be measured with focus on criteria as defined in DIN EN ISO 9241-11 [3]. This international standard defines usability as 'the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use'. Hence, these criteria are also important when choosing an appropriate evaluation method. That means, there are two dimensions for measuring usability, requirements as outcomes of UCD activities on the one hand and effectiveness, efficiency and satisfaction as usability criteria on the other.

In this paper, the authors combined these two dimensions to build a grid that classifies the different evaluation methods, i.e. with regards to their different types of result and the criteria of usability they measure. Usability Tests for example take place late in the development process and can be used to evaluate if 'User Requirements' and 'UI Requirements' are met and if the users can effectively and efficiently fulfil their tasks with the designed solution. Interviews with potential end-users, however, are conducted early in the development process and provide input for 'User Requirements'.

The benefit of this approach is, that evaluation methods can be chosen more precisely based on the type of results (requirements), the relevant usability criteria and their application within the development life cycle.

Benefit

Differentiating the evaluation methods by the types of requirements and the usability criteria effectiveness, efficiency and satisfaction, provides a more refined view of the system's usability. The framework makes it easier and thus more likely that results of all user related activities (e.g. requirements) are evaluated with end-users. It also helps to apply the appropriate methods to obtain results pertaining to a specific usability criterion. Its' independence of a specific UCD process model enhances its' adaptability to different UCD approaches as well as software engineering models.

References

- [1] Boehm, B. (1988). *A Spiral Model of Software Development and Enhancement*. *IEEE Computer*. Vol. 21. pp. 61-72.
- [2] Cockburn, A. (2002). *Agile Software Development*. Addison-Wesley: Boston, MA.
- [3] DIN EN ISO 9241-11 (1998), *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*. International Organization for Standardization, Switzerland.
- [4] DIN EN ISO 13407 (1999). *Human-centered design processes for interactive systems*. CEN - European Committee for Standardization, Brussels.
- [5] Ferre, X (2003). *Integration of Usability Techniques into the Software Development Process*. ICSE'03. Portland, OR.
- [6] KBST (2004). *V-Modell® XT*. Release 1.2.0. Retrieved November 08, 2006 from http://www.kbst.bund.de/cln_011/nn_837872/Content/Standards/V__Modell/vmodell__node.html__nnn=true.
- [7] Zimmermann, D., Groetzbach, L. (2007) *A Requirement Engineering Approach to User-Centered Design*. HCII 2007. Beijing.