10 Collaborative Usability Testing to Facilitate Stakeholder Involvement

Ann L. Fruhling and Gert-Jan de Vreede

Abstract: Stakeholder involvement is an essential part of Value-Based Software Engineering. A critical part of the software engineering life cycle concerns usability testing. System usability refers to the effectiveness, efficiency, and satisfaction with which users can use the system for their relevant tasks. Unfortunately stakeholder involvement in usability testing is costly and challenging to organize. This chapter presents a repeatable collaborative usability testing process supported by a Group Support System that was developed and evaluated in a series of workshops involving a real system. The results show that the collaborative usability testing process facilitates stakeholder involvement through stakeholder expectation management, visualization and tradeoff analysis, prioritization of usability action items, the use of advanced groupware tools, and a simple business case analysis. Furthermore, the process can be considered productive and stakeholders reported substantial levels of satisfaction with it.

Keywords: Usability, usability evaluation, usability testing, collaboration, Group Support System, stakeholder value proposition elicitation, thinkLet, facilitation.

10.1 Introduction

Today, one of the major influencers of most systems’ cost, schedule, and value are software decisions that are inextricably intertwined with system-level decisions (Boehm, 2003). Among these system-level decisions is the system’s usability. System usability is the extent to which a system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specific context of use. System usability is linked to all stages of the software engineering process (Mayhew, 1999). The Usability Engineering Lifecycle, proposed by Mayhew (1999), is a holistic view of usability engineering and illustrates how various usability tasks such as user profile, task analysis, prototyping, and usability evaluation, are integrated into traditional software engineering approaches, agile methods, and object-oriented software engineering (Jacobson et al., 1992).

Systems with a high level of perceived usability are easy to learn and to use (Nielsen and Mack, 1994). System usability concerns various aspects such as the consistency and ease with which users can manipulate and navigate a Web site, the clarity of the interaction, ease of reading, arrangement of information, speed, and layout. Prior research overwhelmingly suggests that system usability is associated with many positive outcomes, such as a reduction in the number of errors, enhanced accuracy, a more positive user attitude toward the system, increased sys-
tem use, and increased user productivity, see e.g., (Lecerof and Paterno, 1998; Nielsen, 1993). Furthermore, a recent study also found that system usability factors are essential elements in conveying the trustworthiness of a Web-based system and do affect users’ perception of trust in personal relationship-based information exchanges (Fruhling and Lee, 2004). In summary, system usability is a key indicator for software engineering success.

A system’s perceived usability level is determined using usability evaluations. Usability evaluations consider the users, the tasks, the equipment, the environment, and the relationships among them (Bevan and Macleod, 1994). Examples of usability evaluation methods include observations, cognitive walk-throughs, interviews and surveys, heuristic evaluations, focus groups, usability testing, and laboratory testing (Nielsen and Mack, 1994). These methods are not mutually exclusive. In fact, using more than one method provides richer analysis and results. As pointed out earlier, usability evaluation can occur throughout the software engineering cycle (Nielsen, 1993). For example, Rubin (1994) classifies three types of usability testing according to the point in the software engineering process:

1. Exploratory testing occurs early in the software engineering process. Its main objective is to evaluate the preliminary design concepts.
2. Assessment testing occurs after the high-level system design has been established. It seeks to evaluate how effectively the system functions have been implemented.
3. Validation testing occurs prior to the release of the system. It compares system performance to benchmark standards in time and effort required to complete a task.

A common reason why system usability breaks down is lack of user input (i.e., a value-oriented shortfall). Collaborative software engineering methods are one way to increase the solicitation of user input in a cost-effective and efficient manner (Dean et al., 1998). System developers often execute usability evaluation methods in a group setting as this may increase stakeholder input and reduce bias; a single stakeholder’s own behavior in using the system may not be representative of that of the general user population. Collaborative usability evaluation can involve different types of stakeholders, often simultaneously – current and target users, usability experts, system designers, and system owners – and, thus, capture valuable insights from all stakeholders throughout the software engineering process.

The purpose of this research was to design a collaborative usability testing process, called e-CUP, and evaluate it in the field. The e-CUP process enables usability practitioners to actively involve different stakeholders and solicit their input and build consensus through synthetic experience techniques, i.e., use scenarios and prototypes. The collaborative usability testing process is an example of how usability practitioners can begin to operationalize stakeholder value proposition elicitation and reconciliation, one of the value-based software engineering approach elements (Boehm, 2003), and in this case, also improve the likelihood that the usability value of the system is considered throughout the software engineering cycle. Boehm suggests five approaches that are most effective for stakeholder value proposition reconciliation which include 1) expectation management, 2)