Chapter 5.29
Integrating Usability, Semiotic, and Software Engineering into a Method for Evaluating User Interfaces

Kenia Sousa
University of Fortaleza, Brazil

Albert Schilling
University of Fortaleza, Brazil

Elizabeth Furtado
University of Fortaleza, Brazil

ABSTRACT

We present artifacts and techniques used for user interface (UI) design and evaluation, performed by professionals from the human-computer interaction (HCI) area of study, covering usability engineering and semiotic engineering, which can assist software engineering (SE) to perform usability tests starting earlier in the process. Tests of various interaction alternatives, produced from these artifacts, are useful to verify if these alternatives are in accordance with users' preferences and constraints, and usability patterns, and can enhance the probability of achieving a more usable and reliable product.

INTRODUCTION

In a software development process (SDP), it is crucial for developers, customers, and users to interact in order to specify, generate, and evaluate the software. From software specification to its delivery, various kinds of tests must be performed, involving aspects such as: functionality, portability, performance, and usability. This work focuses on the context of usability, communicability, and functionality tests (e.g., appropriateness of a chosen interface design alternative to user preferences, consistency to a visual pattern, efficient execution of interactive tasks on interface objects, etc.).

Through our researches on tests in HCI and SE, and through our experiments on their integration in a SDP, we verified that HCI concepts facilitate the UI evaluation work performed by the test team of an interactive system under development. More specifically, by means of UI generation based on HCI models (e.g., task model), it is possible to evaluate the UI earlier (e.g., its functionality), independent of having the entire noninteractive specification ready. Prototypes, for instance, can represent UI design alternatives that may be tested early by HCI experts to verify if they are in accordance with user preferences, usability patterns, and so on.

This work presents a SDP to design and evaluate UIs, based on the integration of concepts, models, and activities of usability, semiotic, and software engineering.

This chapter is structured as follows: The "User-Interface Evaluation" section shows the contribution of each engineering area to UI evaluation; "The Process" section describes the UI design process; "The Evaluation Strategy" section describes the UI evaluation process, showing which concepts are used to perform tests and when they are performed; the "Case Study" section describes the case study in which we designed and evaluated UIs for the Brazilian System for the Digital Television (SBTVD); and, finally, the "Findings and Future Works" section describes findings and future works, and the "Conclusion" section concludes this work.

USER-INTERFACE EVALUATION

In this section, we present concepts and evaluation techniques from usability engineering, software engineering, and semiotic engineering.

Usability Engineering

Usability engineering is a set of activities that ideally take place throughout the lifecycle of the product, with significant activity at the early stages even before the UI has been designed. The need to have multiple usability engineering stages supplementing each other was recognized early in the field, though not always followed in development projects (Gould & Lewis, 1985).

In usability engineering, techniques and methods are defined aiming to assure a high usability level of the interactive UIs. Among them, we emphasize the application of ergonomic criteria in the UI design. Verification of these criteria in designed UIs is called heuristic evaluation, performed by usability experts without user participation. Evaluators examine the IS searching for problems that violate general principles of good UI design, diagnosing problems, obstacles or barriers that users will probably encounter during their interaction. In addition, methods to capture usability requirements attend to user preferences, restrictions, and use-context. A usability requirement can be derived from an interaction restriction; such as if part of the system needs to be implemented for palm-top devices.

The evaluation approaches from usability engineering suggests a structured sequence of evaluations based on "usability inspections methods" and on "usability tests".

Some inspection methods are: (1) heuristic evaluation, verification of usability heuristics (Nielsen, 1993); (2) review of guidelines, verification if the UI is according to a list of usability guidelines (Baranauskas & Rocha, 2003); (3) consistency inspection, verification of the consistency among the UIs related to terminology, color, layout, input and output format, and so on; and (4) cognitive walkthrough, simulation of the user "walking" through the UI to execute typical tasks.

Some usability test methods are: (1) thinking out loud, we request the user to verbalize everything he or she thinks while using the system, and we expect that their thoughts demonstrate how the user interprets each UI item (Lewis, 1982); and