15 PATTERNS OF INTEGRATION: BRINGING USER CENTERED DESIGN INTO THE SOFTWARE DEVELOPMENT LIFECYCLE
Lisa Battle

Lockheed Martin

Abstract

Faced with a need to integrate user-centered methods into existing software development lifecycles, many practitioners lack clear direction and continue to negotiate the scope of their involvement on a project-by-project basis. There are best practices that can be adopted, however. This chapter distills the experiences of many practitioners into a collection of process patterns that describe an evolutionary path towards full integration.

15.1 INTRODUCTION

Despite the increasing recognition of the value of usability and user-centered design in the software industry (Butler, 1996; Trenner and Bawa, 1998) integrating user-centered methods into existing software development lifecycles remains a significant challenge. As discussed in Chapter 2, some organizations claim to be committed to usability but seem to be at a loss as to how to achieve it. The challenge may arise from the fact that the software engineering community already has techniques and tools for
managing the whole development lifecycle, and it is unclear where to integrate user-centered methods (Antunes et al., 2001). Differences in terminology and language may impede communication, since practitioners of usability and human-computer interaction (HCI) typically come from non-engineering disciplines (Ferre, 2003). Or, the techniques of user-centered design (UCD) may appear subjective or disorganized when viewed by developers and analysts who are not familiar with the process (Quesenbery, 2000). There is evidence in Chapter 4 that communication between engineers and HCI professionals remains limited at best.

At the same time, the software industry remains frustrated with the problem of requirements. It has long been recognized that requirements analysis is the most error-prone part of the development process, and errors in requirements not detected at an early stage can lead to expensive system failures (Hofmann and Lehner, 2001; Boehm, 1981). In the 1990s, the field of requirements engineering (RE) gained much more prominence as a discipline, as evidenced by the emergence of several new conferences and journals dedicated to the subject (Nuseibeh and Easterbrook, 2000). RE researchers have proposed a variety of methods for improving requirements elicitation and the communication between users and analysts (Valenti et al., 1998). It has been suggested that these methods would benefit from integration with complementary human-computer interaction approaches (Sutcliffe, 1995).

15.1.1 Representing Practitioner Experiences as Patterns

The question of how to integrate UCD into mainstream software development processes has been discussed by groups of UCD practitioners and software engineers in several recent conferences and workshops (Gulliksen et al., 2003a; Seffah and Forbrig, 2001; Kreitzberg and Quesenbery, 1999; Seffah and Engelberg, 1999). IEEE Software magazine published a special issue on usability engineering in January 2001, suggesting an interest in cross-fertilization. Several case studies have also been published from a practitioner perspective on the experience of introducing UCD into organizations (Wheeler et al., 2003; Anderson et al., 2001; Carlshamre and Rantzner, 2000). However, UCD/HCI practitioners in many organizations continue to negotiate their role and the scope of their involvement on a project-by-project basis. The need for clearer direction to guide practitioners towards strategies for integration prompted our workshop at the Usability Professionals’ Association (UPA) Conference (Battle et al., 2003). In the workshop, UCD practitioners discussed their experiences and shared examples of how UCD was integrated into their organizations’ software development lifecycles.

This chapter is based on the ideas generated during that workshop, a written survey of usability practitioners conducted at the same conference, feedback received from local UPA chapters in response to presentations of this material, and lessons learned from the author’s experience. The information is distilled into process patterns that describe best practices for integrating user-centered design with the software development lifecycle. Process patterns are derived from the idea of design patterns originally introduced in architecture by Alexander et al., 1977, and adapted more recently for user interface and software design. Process patterns draw on combined experiences to describe a proven, successful approach or series of