IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB8283

Chapter VII

Designing for Changing Work and Business Practices

Yvonne Dittrich and Olle Lindeberg
Blekinge Institute of Technology, Sweden

Changes in the use context of computer programs require flexibility and adaptability of the software itself. But, how do you design for change in rapidly changing environments? Based on a case study, the chapter argues that it is not enough to only relate to the use context. Taking also technical and developmental contexts and how they relate to the use context into consideration, allows for light-weight solutions for adaptable special purpose software. The conceptualisation of design as "artful integration" of different contexts provides a starting point for a systematic identification of relevant contexts and evaluation of design alternatives in relation to these contexts.

INTRODUCTION

As the use of computers spreads, software is used to support more and more unstructured business and work practices. Work and business practices—espe-

cially if related to rapidly developing technology such as telecommunications—are under additional pressure to change. The software supporting them must also change. It is no wonder that more effort goes into maintaining software than developing it (Nosek & Palvia, 1990). One way to alleviate the problem is to make software adaptable, i.e., easy to modify with or without programming in accordance with certain anticipated types of changes in requirements.

In this chapter we report from a cooperative project with a telecommunication supplier and a small software developing company.[1] The aim of the project was to develop a software system for a rapidly changing business area. Different techniques to provide for runtime flexibility and to support changes were explored, evaluated and partially implemented. We were involved not only with the development and implementation of the system but also with the interaction with the business unit and future users, the focus being on the design of the software as well as its testing and initial application. We argue that a good solution capable of providing adaptability depends not only on the use and business context but also on the technical and development contexts as well as on the relation between them.

Finally, we summarise our experiences and reflections on the project by giving some more general implications for the development of systems when change can be anticipated. Developing adaptable software in commercial settings requires not only more sophisticated technical design but must adjust the technical possibilities to the situation at hand. The resulting design can be described as an artful integration of technical possibilities and the use, development and technical contexts.

SOFTWARE EVOLUTION AND CHANGING USE CONTEXTS

The issue of a changing use context and growing pressure for change of the software supporting is not new. Already in 1980, Lehmann argued that every program that is embedded into a social context produces its own pressure for further development; it changes the embedding practices. Lehmann formulated the "laws of software evolution" and explained why maintenance efforts do not decrease once delivery has taken place, but may even increase. This issue has been addressed in very different ways in different research communities.

In software engineering, iterative process models have been developed that use prototyping to explore uncertainties in requirements and aim to produce a better fitting application (Boehm, 1988). Evolutionary project models such as STEPS provide a frame for the codevelopment of work practices and technology as a learning process through which developers and users together design and develop the software and its use (Floyd, Reisin, & Schmidt, 1989). In this way, the mutual