Chapter 5
Mutual Development: The Software Engineering Context of End-User Development

Anders I. Mørch
University of Oslo, Norway

Renate Andersen
University of Oslo, Norway

ABSTRACT

The article presents and analyzes data from a case study in customer-initiated software product development. We have observed and participated in system development activities in a commercial software house (company) over a period of two years. The company produces project-planning tools for the oil and gas industry, and relies on interaction with customers for further development of its products. Our main research question is how customers and professional developers engage in mutual development mediated by shared software tools (products and support systems). We have used interviews with developers and customers as our main source of data, and identified the activities (from use to development) where customers have contributed to development. We analyze our findings in terms of co-configuration, meta-design and modding to name and compare the various stages of development (adaptation, generalization, improvement request, specialization, and tailoring).

INTRODUCTION

The goal of the research reported here is to identify areas where end-user development (EUD) and professional software development meet and interact. We have observed and participated in development activities in a commercial software house (referred to as company) over a period of two years. We propose a model of the activities, which we refer to as mutual development. The model consists of the 5 sub-processes, which connects EUD and professional development.
Motivation

There are two levels of software development: specific and general, represented by the activities of amateur (end-user) and professional developers, respectively. When they interact in a mutually beneficial way a new opportunity for innovation emerges that extends the boundaries of what can be accomplished with in-house driven innovation (Victor & Boynton, 1998). We call this model “inside-out” and the model advocated in this chapter “outside-in.” Inside-out innovation is initiated by professional developers and technology managers, that is, technological innovation, whereas outside-in is driven by external events, a type of innovation that originates outside professional communities by end users (Fischer, 2002) and customers (von Hippel, 2005). They are amateur (non-professional) developers who act on behalf of specific needs and by creating ad hoc modifications to “finished” products. The mutual dependency between technological and user-driven innovation can be exemplified by the development of Levi’s blue jeans. In 1873 Jacob Davis and Levi Strauss invented to rivet (copper nail) the pocket corners on men’s working pants to make them stronger. The two men were granted a patent on this, which is widely recognized as the invention of the blue jean. It is an example of technological innovation. Later on, in 1937 the rivets (copper nails) were removed on the back pockets (first sewn so that they were hidden, and during the Second World War removed to save metal). The hiding and removal of the rivets were in response to consumers who complained that the rivets scratched furniture and saddles. This was a user-driven innovation triggered by various external events that modified a technological invention into a more useful design (Levi Strauss & Co., 2009). It serves as a motivating example for the topic of this article, which is about mutual development: the integration of end-user development and professional software development.

Mutual development is based on a technique for participatory design (PD) known as mutual learning (Bratteteig, 1997). Mutual learning is a technique for users to learn about software design from professional developers, and developers learning the professional (workplace) language of domain-expert users. Mutual development takes this a step further in that the shared knowledge established during mutual learning serves as platform for end-user development. The goal is not only to support collaboration between developers and users (as in PD), but also to empower them with new tools and new forms of organization. This is arguably easier when there are two distinct levels of development, one for software engineers and another for end-user developers. Rich communication channels to connect them can stimulate formalizing ad hoc modifications into software artifacts, and a good process of collaborative design can transform an end-user developed solution into a commercial product feature (Mørch, Nygård, & Ludvigsen, 2009). Users will be increasingly motivated to participate in end-user development when the activity has impact beyond a local solution.

The Case

The company is engaged in commercial software development in the area of project planning and management and provides consultancy services in using its tools. At present, the company employs 25–30 people, but they intend to grow and is concurrently expanding their staff and searching for new markets. In fact, the company is recognized as a major player in the business of project planning tools. They have several hundred customers and they have long-term commitments with many of them. One of their recent products is an add-on to Microsoft Project.

The main market has been the Nordic oil and gas industry. To expand into new markets, particularly building and construction, the company has started to modify and improve its knowledge man-
Related Content

User Perceptions and Groupware Use
[www.igi-global.com/chapter/user-perceptions-groupware-use/18228?camid=4v1a](www.igi-global.com/chapter/user-perceptions-groupware-use/18228?camid=4v1a)

Understanding and Evaluating Source Expertise in an Evolving Media Environment
[www.igi-global.com/chapter/understanding-evaluating-source-expertise-evolving/69752?camid=4v1a](www.igi-global.com/chapter/understanding-evaluating-source-expertise-evolving/69752?camid=4v1a)

When Technology Does Not Support Learning: Conflicts Between Epistemological Beliefs and Technology Support in Virtual Learning Environments
[www.igi-global.com/chapter/when-technology-does-not-support/18653?camid=4v1a](www.igi-global.com/chapter/when-technology-does-not-support/18653?camid=4v1a)

[www.igi-global.com/article/toward-comprehensive-framework/3789?camid=4v1a](www.igi-global.com/article/toward-comprehensive-framework/3789?camid=4v1a)