

Knowledge Creation Inside and Outside the Enterprise: A Case Study of Customer-Initiated Software Development

Anders I. Mørch, Kathrine A. Nygård, Renate Andersen

InterMedia, University of Oslo

P.O. Box 1161 Blindern, N-0318 Oslo, Norway

anders.morch@intermedia.uio.no, k.a.nygard@uv.uio.no, renae.andersen@ementor.no

ABSTRACT

In this paper we describe results from a case study in customer initiated product development in a software house in Norway. The company develops and sells project-planning tools for the oil and gas industry and provides consultancy services in using these tools. We entered the company in conjunction with an invitation to give advice on their knowledge management practices for internal communication and customer relations. Our findings pertain to how customers and professional developers engage in “mutual development” mediated by shared software tools (products and support systems) that have evolved over long time. We have used interviews as our main source of data, and identified the activities (from use to development) where customers have contributed to development. These activities provide implications for a new generation of evolutionary design support systems.

INTRODUCTION

We have collected and analyzed data in a software house in Norway (referred to as company) for a two years period. The company is engaged in commercial software product development and develops and sells project planning and management tools and provides consultancy services in using these tools. At the present, the company employs 25-30 people, but it is rapidly expanding their staff and search out new marketing share. The main market has been the Nordic oil and gas industry. To expand to new markets, in particular building and construction, the company has started to change and improve its knowledge management practices regarding customer relations. The company’s customer relations rested largely on oral and personal connections. These practices are time consuming and not aligned with the goal of serving a growing market with diverse customers.

The company is known for their adaptive product development philosophy, i.e. close interaction with customers to develop tailor-made products [6]. Customers are encouraged to report problems, innovative use, and local development to the company. This is supported by communication and information sharing tools [5], which started with the telephone, then mail, later extending to a Helpdesk interface, a complex Customer Relationship Management (CRM) system, and most recently a Web 2.0 prototype. Despite the effort involved in developing new

functionally for talking about existing products, such functionality is highly valued by both developers and customers. This benefits the company in two ways, helping to maintain loyal customers and reaching out to new ones.

The first attempt to improve knowledge management practices involved installing a Helpdesk function, but it did not work well (little used). The goal of this system was to allow customers to send email like requests when they needed help with one of the products instead of using the telephone to contact one of the consultants or developers directly. The information could be organized into a database of functional areas and experts in those areas would attend to the requests, as they emerged, somewhat similar to the AnswerGarden family of systems [1]. Instead of first having to go through an arbitrary contact person (as seen by the company), the customer’s request would be first organized by an automated system and the brokerage and trouble shooting time would be taken off the shoulders of the employees. The interface to the Helpdesk system was very simple and a reason why it was not much used. The customers found it more convenient to contact the persons they knew from before, since the time they purchased and installed the project management tool.

The second attempt was an interactive web-based knowledge tool (web portal for short) to be integrated with a recently purchased CRM system that the company planned to be a communication hub between the two national offices, a link between consultants in the field, and to support customer interaction [8]. We entered the company in conjunction with this initiative and were involved in the attempt to build a prototype of the web portal that was tested in the company by its employees [7]. It was built on the help desk idea (easy to use), but using a different technology to leverage a new type of interactive systems that many users are familiar with nowadays (Web 2.0). The project stranded when we were not able to successfully integrate the portal with the CRM tool (a vendor product that was not open for integration with third party tools without extensive debugging and work around).

The partial failure of these two different efforts of supporting knowledge management calls for a deeper analysis, which is broader than the analysis of the use of new software tools within an organization. It is by now clear to us that the tensions between the different

developmental practices could not be solved by one type of solution alone. The problem calls for solutions at different levels, involving both organizational as well as technical components, and combining information sharing support (Web 2.0) with user toolkits (design environments) for end-user development [3]. This combination is what we mean by Enterprise 3.0. It is here (preliminary) developed based on an analysis of data we collected.

We base our analysis on interviews with developers, consultants, and customers, and on data from a video-recorded workshop. The project planning tools serve as boundary objects [9]. By *mutual development* [2] we mean that both professional developers and end users contribute to development as active participants in both design and use. We identify the range of end-user development activities (from use to design) taking place in the interaction between the company's developers and some of their customers.

FINDINGS

There are concurrently two main activities going on that is (more or less) organized by the company. On one hand, the company is in transition to expand to new markets in order to increase revenue. On the other, they want to maintain good relations with existing customers. The two main activities are referred to as *adaptation* and *generalization* [6]. Adaptation is customer-initiated product development that occurs frequently as many small short-term efforts that contribute to incremental change in the products (e.g. improvement requests and tailoring). Generalization is software product line development, which occurs more seldom, but lasts for longer time, and represents major changes for the company (such as new products added to the product family). The relationship between these two levels of development is depicted in Figure 1. Our working hypothesis is that these are the two prioritized software developmental activities that need to be understood in more detail and their contradictions resolved before new knowledge management practices can be successfully implemented in the company.

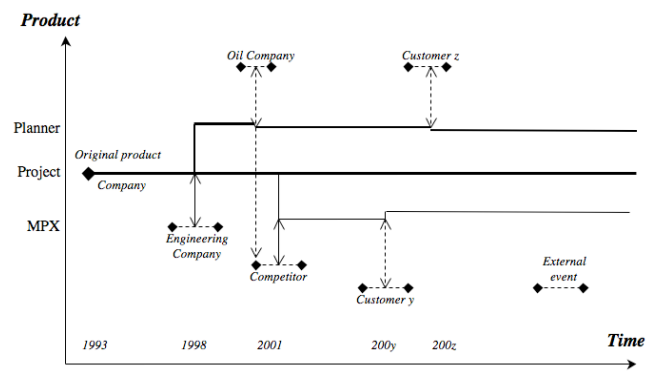


Figure 1. Software product line development in Company depends on interaction with customers and competitors.

The driving forces of software development are *internal and external events*, which initiate the need for changes in

one or more of the products. We focus on modifications to existing products triggered by external events, since our interest is in exploring user involvement in product development and how this can be communicated to professional developers using Web 2.0 and related tools. The changes can be classified as major or minor. By *major changes* we mean changes that lead to a new named product, whereas *minor changes* help to improve and/or maintain the continuation of existing products. The software engineering work required to effectuate the changes are outside the scope of this paper. We have identified customer design activities and developer-customer interaction behavior from our interview data. We summarize the results below.

- It was found that within the interviews there existed some sub-processes of mutual development [2]. They were identified as *Adaptation*, *Generalization*, *Improvement Request*, *Specialization*, and *Tailoring*. They are related as shown in Figure 2, and summarized below as follows:

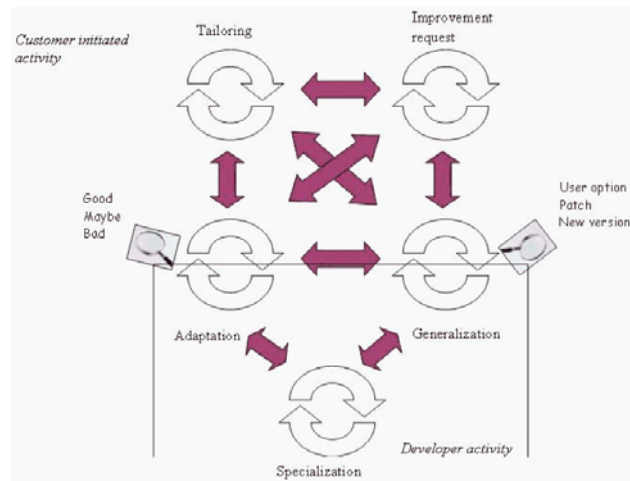


Figure 2. Different stages of mutual development (customer initiated product development): Developer activity and customer-initiated activity co-evolve; the arrows indicate dependencies.

- *Adaptation*: Adaptation is when a customer requests an improvement to an existing product and the company chooses to fulfill the request. It becomes an Adaptation just for this customer. Sometimes, the customer has to pay for this, sometimes not.
- *Generalization*: Generalization occurs when a new version of an existing product is released and is available to all customers.
- *Improvement Requests*: This is when customers request the company for extra functionality, report bugs etc., and is viewed from the customers' perspectives.
- *Specialization*: Specialization is when the professional developers at the company create in-house builds. This could potentially result in new product features, but most

often it entails restructuring the code, perfecting the product, and removing bugs.

- *Tailoring* is about active end users who make adaptations on their own.

The findings reported here have been condensed and depicted in the mutual development model shown in Figure 2. In two related papers the findings are justified by an elaborated analysis of the collected data [2, 6].

DIRECTIONS FOR FURTHER WORK

Our main objective with this position paper is to show how there is mutual development between customers, professional developers mediated by software products in the company we studied. This question points to the components of the process, including two levels of development activity (general and specific) and information sharing between the two levels. This form of mediation is complex due to the dynamic nature of development and interdependency of the activities [4], requiring both organizational and technical components, and information sharing support (Web 2.0) combined with user toolkits (design environments) for end-user development of the project planning tools. In the context of this workshop we would like to explore with the workshop participants what this might mean for Enterprise 3.0.

REFERENCES

1. Ackerman, M. S. and McDonald, D. W. Answer Garden 2: Merging Organizational Memory with Collaborative Help. Proc. CSCW'96. ACM Press (1996), 97-105.
2. Anderson, R., and Mørch, A.I. Mutual Development: A Case Study of Customer-Initiated Product Development. Submitted for review to 2009 conference.
3. Fischer, G., Giaccardi, E. Ye, Y., Sutcliffe, A.G., Mehandjiev, N. Meta-design: A manifesto for end-user development. *Comm. ACM*, 47(9), 33-37 (2004).
4. Fischer, G., Ostwald, J. Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation. In: *Proceedings of the Participatory Design Conference (PDC'02)*, pp. 135-143. ACM Press, New York (2002).
5. Grudin, J. Interactive systems: Bridging the gaps between developers and users. *IEEE Computer* 24 (4), 59-69 (1991).
6. Mørch, A.I., Nygård, K.A. and Ludvigsen, S.R. Adaptation and Generalization in Software Product Development. To appear in Daniels, H. et al (eds.). *Activity Theory in Practice*. Taylor & Francis Books, 2009.
7. Nedic, D., and Olsen, E.A. Customizing an Open Source Web Portal Framework in a Business Context: Integrating Participatory Design with an Agile Approach. Master's thesis, Dept. of Informatics, University of Oslo. Norway (2007)
8. Nygård, K.A. and Mørch, A.I. The Role of Boundary Crossing for Knowledge Advancement in Product Development. *Proc. Int'l Conf. on Computers in Education (ICCE 2007)*, IOS Press (2007), 183-186.
9. Star, S. L., Griesemer, J. R. Institutional Ecology, "Translations" and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology. *Social Studies of Science* 19(3), 387-420 (1989)