LAPPEENRANNAN
TEKNILLINEN YLIOPISTO

Lappeenranta University of Technology
School of Business
Petteri Hilska, 0276862

Bachelors Thesis

SOFTWARE PRODUCT DEVELOPMENT

- the case of Finnish Software SMEs -

Table of Contents

1 INTRODUCTION

The ability to develop new products successfully is critical to any company (Coates et al. 1996; Flint 2002; Kotler and Keller 2006; Lee and O'Connor 2003; McAdam and McClelland 2002; Troy et al. 2001), regardless of their location (Simpson et al. 2002). Considerable amount of research is done on product development (Aramand 2007; Flint 2002; Troy at al. 2001), yet new products continue to fail after their initial launch (Cooper 1994; Flint 2002; Kotler and Keller 2006; Zwikael 2008), in the software industry in particular (Sheremata 2002). New product development can therefore be very risky to the company (Kotler and Keller 2006). Therefore, it is crucial to describe the complex environment of software product development to determine the factors affecting the success of a new software product.

This study focuses on the key factors affecting the product development process of a Finnish software SME. A short overview of new product development is required to understand the concept. This is followed by a more detailed description of the specific characteristics of the software industry. Based on the situation of the case company, very briefly, the effects of partner networks into decision making are presented next. This is followed by the description of the specific characteristics of new software product development and the three available models found in the literature.

The findings of the above concepts are analysed in comparison to the situation of the case company. This is done in two parts, first the current situation of the company is described in detail and this is followed by a more detailed analysis of the key concepts.

1.1 Background

As mentioned above, considerable amount of research is done on new product development. Most of this is done on a generic level, but as stated above, this is needed to understand the concept.

New product development can mean several things. First of all it can mean the creation of radically new product ideas (Kotler and Keller 2006), yet only a small portion of the daily activities of product developers aim at this end result (Conway and McGuinness 1986; Kotler and Keller 2006). For the most parts, new product development is aimed at extending and improving the current product lines (Kotler and Keller 2006).

The success of the new product development process is highly dependant on the company's ability to generate new ideas (Sands 1979). These ideas are then processed within the company to generate tangible products (Sheremata 2002). The process from the preliminary idea to the final product can take many forms. Perhaps quite surprisingly many of the scholars have agreed on a quite similar procedure. As stated above, the process starts with the idea generation phase (Börjesson et al. 2006; Conway and McGuinness 1986; Kotler and Keller 2006; Troy et al. 2001) which is immediately followed by idea screening (Kotler and Keller 2006) or idea evaluation (Börjesson et al. 2006; Troy et al. 2001). Conway and McGuinness (1986) go even a bit further and argue that first of all an idea is not sufficient, it needs to be in a preliminary concept stage before credibility can be gained. Even after the idea is proven to be good, intensive research is required to be sure before the actual development can begin (Conway and McGuinness 1986). Börjesson et al. (2006) and Troy et al. (2001) argue that once the idea is evaluated, it needs to be developed into a product and then launched. Kotler and Keller (2006), however, go into more detail with the process, since while they do agree that these two stages follow idea evaluation, there is also much more. Before the product development can commence the concept needs to be developed and tested, and the marketing strategy needs to be analysed (Kotler and Keller 2006). Then, after the product development is done, yet another testing stage follows, where the product needs to be field tested in the markets

(Kotler and Keller 2006). Only after the success of these tests will the commercialization commence (Kotler and Keller 2006) which is naturally followed by the launch.

Another key notion from the literature is that the new product development needs to be managed in order to be efficient (Conway and McGuinness 1986; Kotler and Keller 2006; Zwikael 2008). This management can mean either top management's support in the form of clear strategy (Conway and McGuinness 1986) or even active participation during the development process (Zwikael 2008). In many cases the focus of management support is misguided, concentrating more on developing single procedures, when in fact they should concentrate on creating realistic and clear objectives for the product developers (Zwikael 2008). Without clear and purposeful objectives development becomes random and too often unprofitable (Kotler and Keller 2006). The key actions for management to conduct consist of: the choice of suitable project managers for product development projects (Kotler and Keller 2006; Zwikael 2008), facilitating the communication between the project managers and top management (Zwikael 2008), creation of measures for projects (Zwikael 2008), management of inter-departemental (Zwikael 2008) or cross-departemental (Conway and McGuinness 1986; Kotler and Keller 2006; Lee et al. 2001; Sheremata 2002) teams for projects and resource allocation for projects (Kotler and Keller 2006; Zwikael 2008).

Organisational structure also affects the success of new product development. Organic organisations are more likely to succeed in their development projects whereas mechanistic organisations are more likely to run their operations efficiently and quickly. (Shereremata 2002).

Kotler and Keller (2006) argue that product development should be customer-driven. Too often, however, the importance of customer understanding is reduced by the emphasis on technical decisions (Lee et al. 2001). Many companies might not even be aware of the importance of customer information, or which type of information to gather (Flint 2002). Therefore, companies should include customers in their development process to make sure that the new improvements are those that are needed in the markets. Customers are also the best target group for testing since they will eventually use the products (Kotler and Keller 2006).

New product development is not an isolated process. The environment has a substantial impact on the development process (Börjesson et al. 2006). The efficiency of new product development is hindered by many external factors including changes in regulations and consumer needs, tightening requirements for the speed and cost of the process and shorter product life cycles (Kotler and Keller 2006). As stated above, the ability to generate new ideas is essential in new product development, therefore in many industries developers are facing a situation where basically all that can be done has already been done with their respective products (Kotler and Keller 2006).

1.2 Research Problems

The main research question for this paper is to describe what are the key elements affecting the software product development in a Finnish software SME.

The sub-questions are:

1 What are the key characteristics of software industry
2 What are the networking benefits and limitations
3 What are the available models for software product development
4 What are the characteristics and limitations of these models

It is firmly believed that the answers to these sub-questions will provide meaningful insight into the complex concept of software product development. The findings of this study will also be useful, for not only the case company, but potentially to other software SMEs which are in the process of improving their new product development process.

1.3 Theoretical Framework

The theoretical framework for this study is as follows:

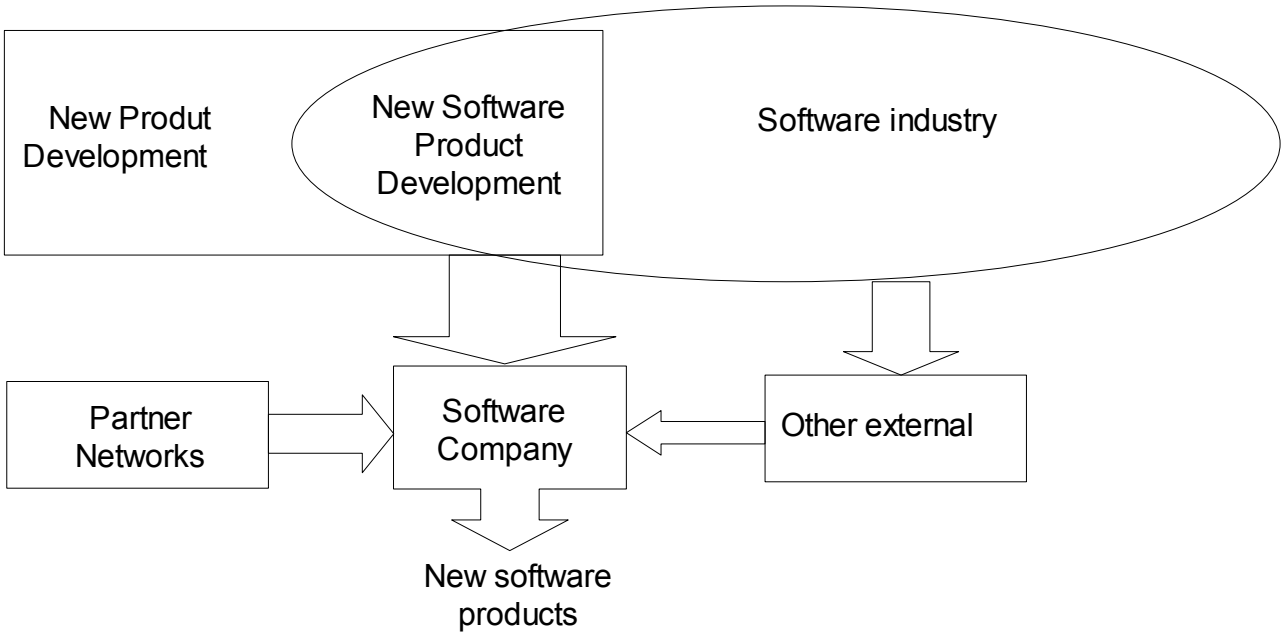


Figure 1. Theoretical Framework

1.4 Delimitations

This research is focused on the software product development process in the small and medium sized enterprises (SMEs). Any models or characteristics, that apply only or mainly to large companies are therefore left out.

Different development models are rated based on their relevance in this environment. Models and definitions for high tech companies are included with minor detail as guidance, since so little is written on the software product development in particular, and since the software industry is a part of the high tech industry.

Previous research done on software industries on specific countries were left out, unless that particular country is Finland. This is done because we acknowledge that culture and

country-specific issues have an impact on software industry and the focus of this study is Finnish SMEs.

2 SOFTWARE INDUSTRY

Software industry consists of the different organisations that operate in the designing, maintenance and publication of the software applications (Aramand 2008). It is one of the fastest growing industries in the world (Aramand 2008; Harris et al. 2007), some even argue that it is the most important industry in the world (Anselmo and Ledgard 2003), yet the record for successful products is rather poor (Anselmo and Ledgard 2003; Sheremata 2002). This added to the high costs of software development – most of the total costs actually appear in the early stages of the development (Sheremata 2002) – causes development managers to seek for more efficient development processes (Sheremata 2002).

Another aspect to consider are the much too regular slipped schedules and cost overruns (Zwikael 2008(which implicate that the software product development process is in a need of management and coordination (Sheremata 2002). Therefore, there is a need for comprehensive study of the complex process of software product development.

2.1 Characteristics

Software industry differs from traditional industries in many ways. First, the user and the customer of a software product may not be the same (Aramand 2007). The user in many cases, especially with Internet software, does not even pay for the use of a particular software (Aramand 2007), which makes it rather difficult to assign revenue on that particular software by the customer. One solution to this has been the pay-per-use -option for the product or service (Gurnani & Karlapalem 2001). This way the customer could allocate revenue, and therefore value for the company, to the product sold by the software company.

Traditional high tech products are described as being state-of-the-art and have a short product life cycle (Aramand 2008). The rapidly changing environment in software industry causes software companies to introduce new applications and upgrades constantly and this shortens the life cycles of the products (Aramand 2008; Cusumano 2007; Harris et al. 2007). It is also much more difficult to estimate consumer needs and evaluate competitors in this fast evolving environment (Antony and Fergusson 2004; Ruokonen 2008; Ruokonen et al. 2008). Market-oriented companies are therefore far better in providing value for their customers since they are better able determine the unmet needs of their customers (Ruokonen 2008). State-of-the-art product is new to the market place (Aramand 2008), but in the case of software products this is not always the best situation. Software usage requires training and a new technology requires new training, therefore many users decide to stay with the existing technology (Aramand 2008).

Software industry in the U.S. consist mainly of small companies (Harris et al. 2007) and it can safely be assumed that this is also the case world-wide. Because of the uniqueness of software products (Antony and Fergusson 2004) the domestic markets are usually rather small. In addition, due to the nature of software products they are easily sold via the Internet. Therefore, many small software companies seek growth and profit from foreign markets, usually entering several markets simultaneously (Ruokonen et al. 2008).

The coming of the Internet revolutionised the software industry in the early 1990s (Aramand 2008). Because of the new opportunities offered by the Internet, new industries started to see the benefits of software applications and the customer base for software companies grew in volume and variety (Aramand 2008). The Internet also increases the competition, since customers can gain access to software providers around the globe, which in turn increases the quality of products and services offered (Aramand 2008). Finally the effects of the Internet have shortened the life cycles even more (Aramand 2008).

2.2 Software Product

Software product is in many ways different from traditional products. The most important difference is the intangibility of software products (Antony and Fergusson 2004). Software products are in many cases customised to customer needs and therefore are unique (Antony and Fergusson 2004; Aramand 2008). Software products are actually rather similar to services and therefore the measures for software quality are different than in traditional product industries. Software products are measure based on their functionality, complexity and quality (Anselmo and Ledgard 2003). Software quality is a reference to the number of defects in the software (Anselmo and Ledgard 2003; Antony and Fergusson 2004), but it can also describe various other measures such as time consummation (Anselmo and Ledgard 2003; Antony and Fergusson 2004). In conclusion, software product quality is more difficult to measure than with traditional products, but without measuring one cannot expect to see any development (Anselmo and Ledgard 2003).

Software applications can be divided into three categories:

1) software projects that are almost entirely customised on the preferences of the customer (Aramand 2008, Ruokonen 2008)
2) software services in which minor adaptation can occur prior to the subscription of the service (Aramand 2008)
3) software products which are standardised products and usually have longer life cycles than software services or projects (Aramand 2008, Ruokonen 2008)

Typically companies offering software projects are operating in closed markets in co-operation with limited customer base, whereas companies offering standardised software products have a much wider customer base, and they tend to operate in an open competitive environment (Ruokonen 2008). Ruokonen et al. (2008) argue that the market has a huge impact on which product strategy the companies choose. According to them, if there is a strong market pull companies tend to follow standardised product strategy and they also tend to gather customer satisfaction data that dictates their business making (Ruokonen et al. 2006). In contrast, if there is an evident technology push the companies

usually offer customised software projects and they require more detailed information about their potential customers – they need to understand them (Ruokonen et al. 2006). Ruokonen (2008) argues that it is much easier to gain customer information in the latter case, because of the interaction between the company and the customer.

Ruokonen (2008) also argues that while in software projects the ownership of the software is transferred to the customer, software products are usually only licenced for use. However, while Aramand (2008) agrees that software projects are the property of the customer, he suggests that software products also become the property of the customer after their purchase. It can very easily be that Ruokonen and Aramand had different kinds of software products in mind while arguing and we also have to notice that while Ruokonen (2008) proposes software products and projects to be the two types of product strategy in the software industry, Aramand (2008) lists three types, software products, services and projects. According to Aramand (2008) software services are more like Ruokonen's software products as Aramand argues that software services are those that customers can use and customize but they do not own the actual software.

Regardless of the definition, software products are rarely finished, since continuous upgrades and modifications are made (Aramand 2008) in order to address the new-found needs of the customers. In fact, software products typically have a life cycle of 12 to 18 months (Aramand 2008). Because of this constant adaptation and short life cycle, software products can actually avoid the decline phase altogether (Aramand 2008).

3 PARTNER NETWORKS

Sometimes companies engage in different kind of networks, usually to speed up the growth of the company. There are many levels of networking from outsourcing to joint development. Depending on the needs of the company they need to decide on the appropriate level of networking. (Kulmala and Uusi-Rauva 2005) According to Kotler and Keller (2006) network members need to be trained so that they are able to perform according to the values and expectations of the company.

However, the company managers should carefully analyse their needs before engaging in networks. To operate efficiently in networks requires alteration within the company. Companies in networks require open communication about the needs of the companies in order for the network to bring any benefits to its participants. (Kulmala and Uusi-Rauva 2005) Kotler and Keller (2006) advice companies to evaluate the members of their networks occasionally. Some members might not be performing according to the standards appreciated by the company (Kotler and Keller 2006). But if the networking succeeds it can bring substantial cost reductions and new information. This leads to higher growth estimations as stated above. (Kulmala and Uusi-Rauva 2005).

4 NEW SOFTWARE PRODUCT DEVELOPMENT

Software product development can be viewed as an active process to find solutions for problems (Sheremata 2002). As described above, the environment is constantly changing and evolving, hence the problems are constantly changing. The software product managers need to decide which problems to analyse and focus on, since the magnitude of opportunities can be overwhelming (Sheremata 2002).

4.1 Characteristics

Software development process can be divided into phases as shown in the figure 2. The first phase is the analysis of the need for a new software product (Aramand 2008) – the needs origins may be internal or external. The technical department is in charge of the following two phases, creation of the software architecture and programming the actual product (Aramand 2008). The testing phase is usually done by various people including the programmers, lead customers and the marketing department (Aramand 2008). The idea of this phase is to test the product for performance and quality (Aramand 2008). The final phase is the final product modified based on the results of the testing phase (Aramand 2008)

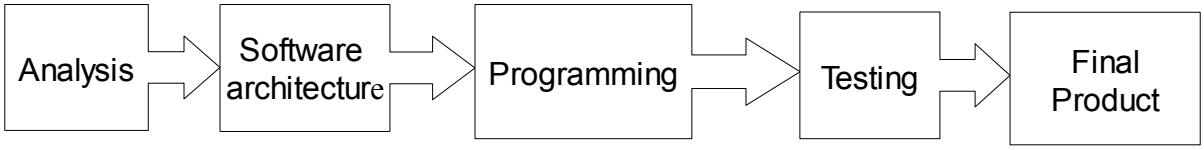


Figure 2. Software product development process

As described above, software product development is a continuous process. After a particular software product is sold, it requires maintenance. Maintenance in this context refers to modifications made by the request of the user or the customer (Aramand 2008). Software enhancement, however, refers to changes made to the existing product by the developers based on their anticipations on what may be the needs in the future (Aramand 2008). In either case, the changes made need to bring added value to the product for the user, or there will be no reason for the user to upgrade his product (Aramand 2008).

When compared to the traditional phases of new product development (Börjesson et al. 2001; Conway and McGuinness 1986; Kotler and Keller 2006; Troy et al. 2001) we can see that the phases of new software product development are not that different. There is slightly more emphasis on the technical part of product development but this is can be explained by the technical nature of software products. Also the emphasis on after-sales support and maintenance in product development is much higher in software product development than in the traditional due to the volatile environment.

In addition, Sheremata (2002) successful product development is dependant of two factors: information and integration. The more information the developers have and the more integrated the process is, the better quality products will they will produce (Aramand 2008).

The available information is high if the process is decentralized and the information flows freely inside the development team (Sheremata 2002). The process is well integrated when development team participants come from various functions and have direct contact with each other (Sheremata 2002). The process also needs to be managed through mini-milestones (Sheremata 2002).

The timing of information gathering is also crucial for the success. Developers are advised to start the process as early as possible (Sheremata 2002). This ensures that problems are recognised at a very early stage so the problem solving can begin earlier and can be conducted more efficiently (Sheremata 2002)

4.2 Development Models

The models and/or frameworks included in this study are those that are designed for small or medium -sized software companies. The models included can be found in the table 1.

Table 1. Models or frameworks included

Model	Limitation	Description
4CC Framework	Framework for small IT companies	Holistic view on software product development
Whitewater process	Process designed for small IT companies	Idea – micro releases – tech support – feedback – platform
Launch – Re-launch	Model for high tech products in general	Difference between the visionaries and the mainstream market

4.2.1 4CC Framework

The framework created by Rautiainen, Lassenius and Sulonen (2002) is designed for small IT companies. Development managers need to balance between business and development aspects and they need to have a high degree of control while also having flexibility at work (Rautiainen et al. 2002). This is explained later with further detail.

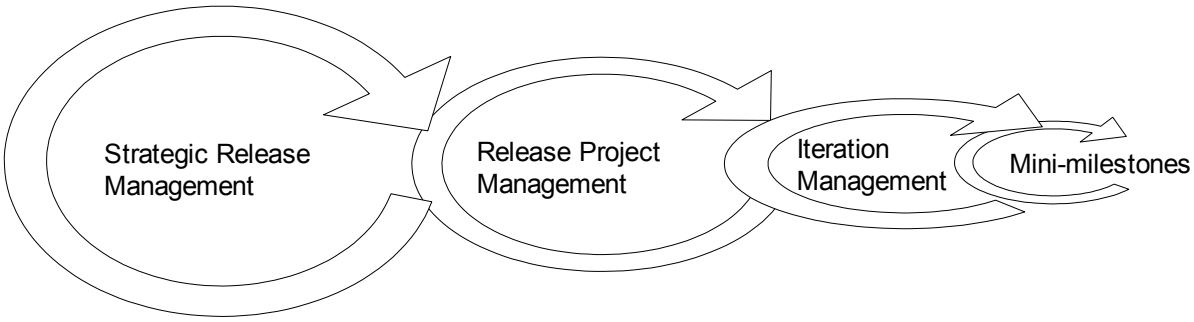


Figure 3. The four cycles of control framework adapted from Rautiainen, Lassenius & Sulonen (2002)

The framework has at the same time both a long-term and short-term view on the product development (Rautiainen et al. 2002). The main benefit of using the framework is to develop understanding of how the product development can be done and also to have control over the different stages (Rautiainen et al. 2002). The main purpose of this framework is to produce new products in shorter cycles and get feedback from customers at a very early stage (Rautiainen et al. 2002).

The four cycles of control are 1) Strategic Release Management, 2) Release Project Management, 3) Iteration Management and 4) Mini-milestones (Rautiainen et al. 2002). The first cycle, Strategic Release Management, has a general and long-term view over product development. The purpose of this stage is to set the general guidelines for product development (Rautiainen et al. 2002): where to go, what to research. All the key stakeholders participate in this stage to have their input on how the company should operate. Most of the major decisions are made at this stage (Rautiainen et al. 2002) to facilitate the following stages.

The second stage or cycle is the Release Project Management (see figure 3). The key decisions to be made in this stage are the plans for version releases. With the guidelines from cycle 1, managers need to decide which way to go with the product (Rautiainen et al. 2002). Each individual project is treated independently from other projects (Rautiainen et al. 2002). The main purpose is to create a baseline from which continue (Rautiainen et al. 2002). After this, managers will give feedback on the guidelines given to their superiors responsible for the previous cycle as well as plan the iteration stage (Rautiainen et al. 2002).

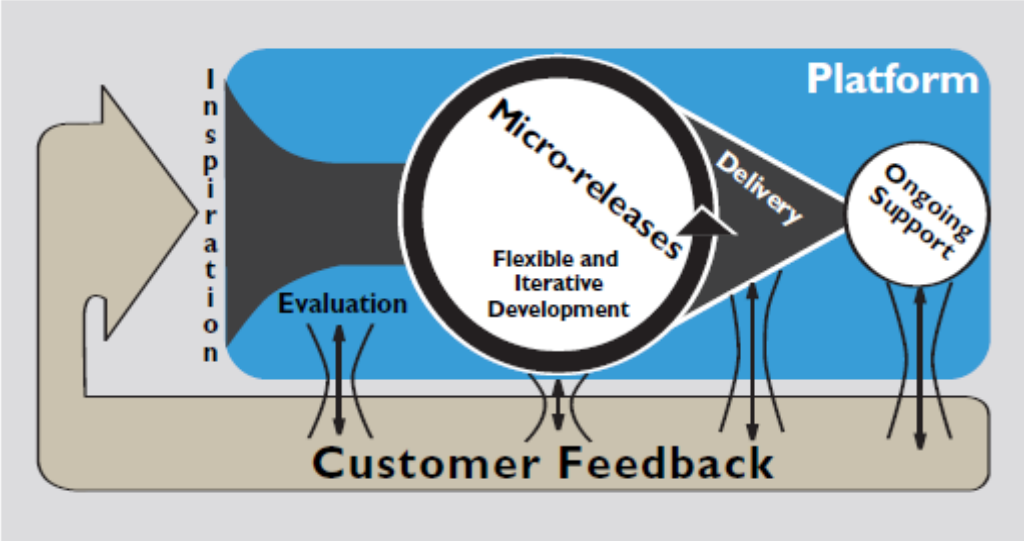
During the third cycle managers will develop a stable, working product on the baseline given (Rautiainen et al. 2002). The instructions are rather loose so that managers have some freedom to experiment, as long as the end result is a working product (Rautiainen et al. 2002). The main idea of this stage is to create a detailed plan and timetable for mini-milestones and also to give feedback on the instructions given from previous cycle (Rautiainen et al. 2002).

The last cycle are the mini-milestones (see figure 3.) In this stage the product is built with daily build tests by individuals and teams (Rautiainen et al. 2002). Feedback is given on the progress and success of these tests to the previous cycle (Rautiainen et al. 2002). Since the tests are done daily, any problems in the product can be seen at a very early stage (Rautiainen et al. 2002) and be removed. Overall the whole process is very iterative and a certain degree of freedom is kept throughout the process.

4.2.2 The Whitewater Process

The Whitewater Process is most suitable to small IT businesses which lack the economies of scale that larger companies have (Harris et al. 2007). Therefore they cannot use the same traditional methods used by the larger companies and they need to rely on a more product-oriented strategy (Harris et al. 2007)

The study conducted by Harris, Aebischer and Klaus (2007) on three small IT business in Florida revealed some common methods in their product development and these methods form the basis of the whitewater process. The five components of the whitewater process – inspiration and evaluation, micro-releases, delivery and high-touch support, feedback and control, and technology platform – can be seen in the figure 4.



Ficture 4. The Whitewater software development process (Harris et al. 2007)

The product development should start with a phase of idea generation, where the ideas can arise from customers, within the company or from a market scan (Harris et al. 2007). The customers are the main measurement when evaluating the ideas, but a small company cannot agree on all the demands from the customers (Harris et al. 2007). Small companies cannot take substantial risks either, because of their low economies of scale (Harris et al. 2007), so to minimize the risks they need to prioritize, limit the scope of individual developments and listen to the key customers (Harris et al. 2007).

As mentioned above, software products in general have a short life cycle. Therefore software companies need to be able to produce new releases to the markets quite often (Harris et al. 2007). Even though this may seem quite a task for a small company, when taken into consideration that they do not include major changes in each release, it is relatively easy to produce an upgrade.

Even though their strategy is usually more product-oriented, the customer is still the key. Small software companies are committed with high level of tech support for their customers (Harris et al. 2007). Since they usually have less customers than bigger companies, this task is not as difficult to perform as would be assumed. In addition, commitment to the tech support enhances the relationship which is vital to the survival of the company. It also may trigger new idea generation phases for the technicians, since

they are able to see the product in use and to get direct feedback from the customers (Harris et al. 2007).

Another key notion from the study is that small IT companies are not pioneers and generally not even early adopters of new technologies (Harris et al. 2007). Their development strategy is to build one platform to use and make modifications on that (Harris et al. 2007). This approach allows effective business even with low economies of scale. In addition, instead of having the latest technologies and features, small IT companies have what their loyal customers are looking for (Harris et al. 2007).

The study was conducted on very a specific environment and while it may contain useful information for development managers, it may not be usable in every situation. The conditions of the study are as follows: the study was done on small companies, with small development teams; all the products were available through the Internet; none of the companies had more than few hundred customers (Harris et al. 2007).

4.2.3 Launch – Re-launch

Easingwood and Harrington (2002) describe a very accurate model on the product launch. According to the model, a new product launch needs to first please the visionaries before it can be adopted by the main market (Easingwood and Harrington 2002). Yet, even a very successful launch will not guarantee that the product will eventually be adopted by the majority (Easingwood & Harrington 2002). The model can be seen in figure 5.

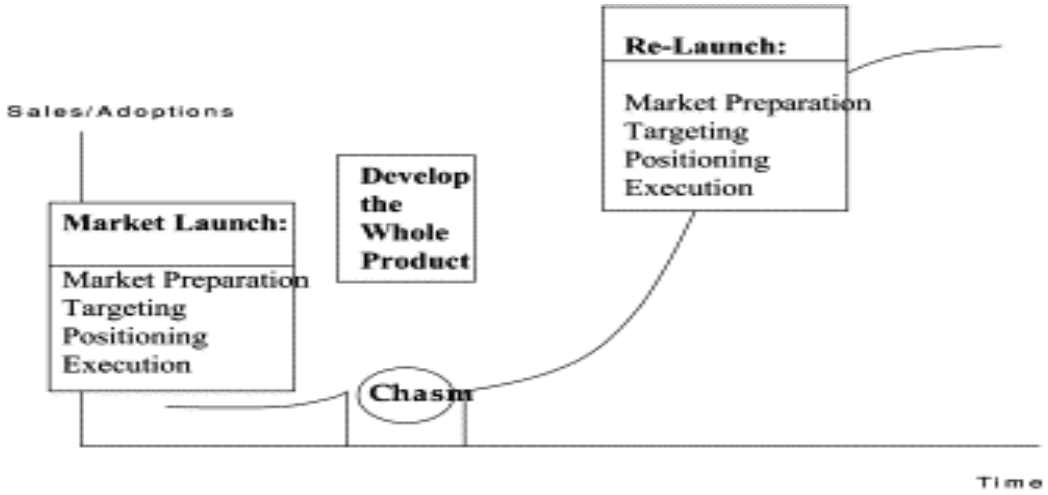


Figure 5. Launch, whole product, re-launch (Easingwood & Harrington 2002)

According to Easingwood and Harrington (2002) much of the actual development takes place only after the initial launch. During the first launch the marketers need to prepare the markets for the new product, find the right segment and position their product accordingly (Easingwood and Harrington 2002). As stated above the initial segment is usually the visionaries. Therefore the product actually needs to be very well specified and a working model too. But one of the key characteristics of the visionaries is that they, in many case, wish to participate in the development of the product, often suggesting valuable additions to the product (Easingwood and Harrington 2002).

As stated above, much of the development is done after the initial launch, once the product has entered the chasm. The purpose of this development is to prepare the product for the mass markets, since the mainstream markets appreciate stability and practical functions (Easingwood and Harrington 2002). The norm is that every product needs to enter the chasm (Easingwood and Harrington 2002).

The model suggests that after the period in the chasm the product needs to be re-launched. The actions to be conducted are the same as in the original launch, yet the purposes might differ slightly. (Easingwood and Harrington 2002)

4.3 Synthesis of the Models

The three chosen three models have slightly different approach to software new product development. The 4CC framework (Rautiainen et al. 2002) emphasises the importance of control on the process whereas the Whitewater Process (Harris et al. 2007) argues that the customer is the key and small software companies need to follow incremental development model. Easingwood and Harrington's (2002) Launch – Re-launch model however argues the importance of marketing actions on the success of the new product. The following is a synthesis on the ideas found in the three models.

Both the 4CC framework (Rautiainen et al. 2002) and the Whitewater Process model (Harris et al. 2007) suggest that software product development should be done in small incremental steps. The customer is seen as the key when deciding which new functions are included (Harris et al. 2007). But while Whitewater Process is mainly focused on the iteration cycle of product development (Harris et al. 2007), the 4CC framework also emphasises the importance of long-term planning and clear strategy (Rautiainen et al. 2002). Harris et al. (2007) also recommend their model to be used only in small software companies, they do not guarantee that the model is applicable in other situations, whereas Rautiainen et al. (2002) while stating that this model is best used with small software companies do not state similar restriction on their framework.

Easingwood and Harrington's (2002) Launch – Re-launch model also views product development as a continuous process. By repositioning their products at regular intervalls companies can actually avoid the decline stage altogether. However, it must be noted that companies need to be aware of the existence of the chasm, since if not prepared for, it can really prove to be fatal for the company (Easingwood and Harrington 2002).

5 COMPANY X

The case company is a medium-size Finnish software company whose headquarters is located in Helsinki, but whose operations cover the entire globe. In fact, 2/3 of their turnover comes from abroad. Most of their revenue is made out of licence sales and maintenance services.

The following five chapters describe the various new product development related concepts within the case company. These concepts include the objective of the new product development process, the process itself, the monitoring procedures, the possible restrictions and a short description the the launch procedures as well.

The following findings are based on the interview that was carried out with the marketing manager of the company (see appendices 3 (Finnish) and 4 (translated into English)). The plan was to have the first interview with the marketing manager to determine whether there was need for additional interviews. The information gained from the marketing manager was such that no additional interviews were seen necessary.

5.1 Purpose of New Product Development

The marketing manager of the case company stated that the new product development is indeed very important, in fact they spend 21% of their turnover on product development. The company has two software products of their own on sale and all of the efforts of new product development strive to improve these two products.

Currently the company does not engage in innovative product development. All of their development efforts are placed on product improvement and problem solving.

5.2 Product Development Process

The duration of the product development process is typically from nine to twelve months. Most of the new features are on hold for this time, since the company tries to avoid introducing new feature between releases. However, the company relases smaller service packs once a month or every other month. These service packs are upgrades on the current (or any previous) version that solve some or all of the problems encountered. The relase of a service pack is triggered by the encounter of a critical problem within any of the available product versions.

If any developer wishes to make customisations between releases this is made possible by the interfaces. This is not recommended, however, since it requires a lot of time and resources.

The following chapters will describe the key stages in the product development process, the participants and in more depth the role of the tech support.

5.2.1 Phases

The marketing manager stated that there are 7 stages in their product development concerning a single feature. These are as follows:

1. Identification of new features
2. Operational specifications
3. Include/Exclude
4. Technical planning
5. Coding – code review – testing
6. Working feature
7. Inclusion in the next release

The key idea throughout the whole process is to make sure that the end result is something that is needed. All of the potential ideas are stored in a bugs&features database before specifications are made. Based on these specifications a workload estimation is made which tells the product managers whether the feature can be included in the next release or not. Expertise is used during the technical planning to determine what needs to be done. Product development teams do the actual coding, but inside the teams peer evaluation is implemented, since the three phases – coding, code review and testing – are done by different people.

The above phases are for single features as stated. Concerning the actual software product, testing starts a few months prior to the release. At this stage all the intended features are included and the field testing commences. The case company uses its partners very often at this stage, since the partners have access to customer databases and models and can therefore test whether the product would work withing their systems.

5.2.2 Key Participants

The most important factor in product development is the product management. They are responsible for the direction where product development is going, they acquire and analyse the required information for new ideas and they ultimately make the go-no go -decision with new features. The road mapping they do for product development usually spans over 3 years.

As stated above, the product management filters the information available and decides which ideas to proceed. The most important source for new ideas are the customers. The case company views itself as market-oriented, meaning that they try to predict the needs of the markets. However, 80% of their new features are based on customer requests. The technological knowledge within the company has a minor role on product development, it sets the limitations to what can be done with reasonable effort. The partners are also an important source for new ideas, since quite often they gather ideas from the customers and forward that information to the case company. Other sources for new ideas include top

management, company strategy and competitors.

Generation of new ideas is only the first stage of new product development and even though it is probably the most crucial stage, much work is done after the initial stages. The actual coding and testing is conducted in the product development unit consisting of around twenty people. The company has quite efficient automatic testing environments and the the automatic build-tests are done three times a day. It can be said that testing the functionality of the product is also very important.

5.2.3 Role of Tech Support

The role of tech support is more of problem solving, even though much of this work is outsourced. All in all, finding solutions to problems with customers is rather easy, because of the nature of most of the problems. The fact that not all use the same version causes some additional work for tech support, since they need to be aware of several differences between different versions.

The customer care support is a part of tech support within the company, yet it is not considered a part of product development. The customer care is also a general email-address for the company that customers and partners use to send in ideas and suggestions and questions. Therefore customer care can be a valuable source of new ideas and therefore tech support as an entity has an important effect on new product development.

This was the case a few years ago. Currently the customer care is not providing as much information as it used to. Own people visiting the customers, seeing the product in use are producing more relevant ideas than the emails from customers and partners.

5.3 Process Management

New product development is very independent work at the case company. There are certain measures that are calculated and used, but the management believes that there is no need for supervision or monitoring. All the work done on the features is recorded in a database and there are bonus schemes related to the efficiency so the workforce is motivated to be efficient. Bugs in the software are also listed and they are categorized.

The company also uses customer satisfaction as a measurement. They conduct once a year a customer satisfaction survey, and based on the results can monitor whether they are producing those features that their customers are looking for. Also the length of the maintenance service the customers are using informs the management on whether the customers place value on the upgrades provided.

The most difficult part to monitor is that of product management. There is no record on which features are chosen and why, so it is not possible to monitor whether the management works efficiently or not.

5.4 Restrictions

There are two key factors causing restrictions for new product development: the company strategy and the international partner network.

The company strategy provides the long-term direction for the product development. The product management needs to look into the strategy to see where to go with the development, which customer requests to include and which to drop out. This is done so that the development is purposeful and profitable in long-term.

The international partner network causes restriction and provided opportunities. First of all the partner network is used internationally, so the company has relatively little direct contact with their international customers. The company has relatively many partners who do not do much business per partner, so therefore product sales are not of their key interest. The main purpose of the partner network is to provide the consultancy layer between the customers and the company. This in turn requires that the product is easy to use, since the partners do not posses the skills to use complex systems. However, this can be seen as an advantage too, since the customers appreciate easy-to-use software as well. The partners participate in the localization process since they have the better knowledge of the local needs as well as the local language (a translation to arabic has been made). In addition, as stated above, the partners themselves can be a valuable source for new ideas and the company uses them as a testing environment for the new release.

5.5 Releases

As stated above, the cycle for releases is from nine to twelve months and for the service backs about one and a half months. The service packs are included in the maintenance pack, so they are never sold nor promoted and no price or value is placed on them.

The company wishes to view its business as a continuity with the customers. Therefore they do not allocate any value for the releases, even when the customers order their product, they do not state which version they are acquiring. Whenever they release a service pack or bigger version release, it is always available for all. They can be included in the operating systems of the customers when they wish and usually are included only when the customers confronts a problem with the product. Even though the company keeps record on which versions are in use, this is for customer satisfaction and new idea generation efficiency purposes, not to calculate revenue for certain version.

6 ANALYSIS OF CURRENT PROCESS

New product development is seen as a critical task within the company. However the end results have been only incremental improvements, the company has not launched any new innovative products in ten years.

The process is managed both in long-term perspective as well as with short-term objectives. The top management designs the strategy which sets the guidelines for the product development. Product management in turn bases their decision making on the strategy and sets objectives for product developers.

6.1 Development process

The duration of the product development process is slightly shorter than the average in the industry, being from nine to twelve months, but since the new release does not immediately out-date the older versions, the products might have substantially longer product life cycles.

The stages of new product development are quite similar to those of the models proposed. The emphasis on technical planning and coding is understandable because of the technical nature of software products. Software products with many defects will undermine the reputation of the company and customers will eventually leave. Therefore it is critical to test that the features and the product itself functions as well as possible. It is also vital to test whether the product development has kept clear focus on what the product was meant to be. Therefore the field testing at the end is also crucial and according to the models. The company also has a very effective maintenance service for their customers, however it is seen only from the problem solving perspective. The company does not believe that the maintenance improves their relationship with the customer and the generation of new ideas is only in minor role.

The company values feedback and believes it to be one of the main sources for new

development ideas. However the other part, control, is left to minimum. The work done by programmers is saved on different databases that are monitored, bugs in the software are listed and classified and customer information is also analysed and stored. Much monitoring is actually done, but there is no apparent need for close supervision of the programmers. The company believes the results will state whether the work done is sufficient enough. However, in some cases waiting for the response from customers might prove to be fatal, but it seems that this style of management is providing results for the company.

The company only sells two software products which consist of several similar software components. These two products offered are licenced for use and the choice of version and improvements is in the hands of the customers. Customisations based on a single customer are left to minimum, but if needed it can be done via the interfaces. In fact many of their customers have customised the software to better suit their business environment. This in turn increases the workload and knowledge requirements of maintenance.

6.2 Participants

The customer is seen as the most important factor in product design and development. The company does not wish to view itself as customer-driven, however, but rather as market-driven. However, 80% of their new features are based on customer requests. The market-driven mentality might refer to the mindset of the product managers who make the final decisions on which features to include in the next release and which to leave out. Therefore much monitoring and critical evaluation is done at this stage, even though it was not clearly stated as one of the stages. It was pointed out, however, that monitoring the product management is the most difficult task because their actions are not saved in any of the databases.

Partner network is very crucial in the development process. As stated above they get most of the turnover from international markets and they operate in these markets through their partners. The company does not have the resources to cater all the markets by themselves and they use their partners for consulting their customers. Partner network is

also used to test the new releases. Because of the size of their partner network the company does not have the time or the resources to fully educate their partners on how to use complex software. Therefore they have decided to keep the complexity of their products to the minimum so that the new releases are easily adapted by their partners and in fact eventually also by their customers.

6.3 Marketing decisions

The company has no control on what versions the customers are using or when they decide to upgrade. The only control and measure the company has is the length of maintenance support their customers are buying. This is their justification for new product development, as long as customer value the improvements made by the company, new product development remains profitable. Therefore new product development efforts can only be measured based on internal efficiency, not on the revenue gained from certain activity.

Most likely due to the short duration of their product development, the company has managed to avoid the chasm quite efficiently. Actually they do have differences between customers, where some are more keen to try new versions or service packs just because they improve some functions slightly, the others wait until they pretty much have to upgrade due to critical problem in the software. Therefore the company actually has different versions in use all the time and some are being used by the “visionaries” in their field and others by the “mainstream” of their customer base. Since the company does not keep record on which versions are in use it is impossible to tell the relative difference between these two groups, but then again there seems to be no need for this division. The company improves those products that are in use and that need to be improved.

7 CONCLUSION

New product development is one of the most crucial processes within any company, but especially so in the high technology industries. The ability to create new products or improve the existing product range will determine whether the company will remain profitable or not.

Much research is done on new product development from various perspectives. When comparing new product development in the traditional industries and in the software industries, the processes are surprisingly similar. The major difference is the emphasis on technical planning and maintenance in the software industry, in contrast to marketing efforts in the traditional industries. Also, software products differ from other products because of their intangibility and, in some cases, their high level of customisation. Therefore, software products are rather unique compared to other products which are quite standard. Other than those differences software product development goes through the same major stages from idea generation to launch. The customer is the single most important source for new ideas in software industry as well as in the traditional industries.

Three models for software development in software SMEs were included in this study. The three have different perspective on product development: the 4CC framework emphasises the management aspect, the Whitewater Process model has a holistic view on the process itself and the Launch – Re-launch concentrates more on the launch phase. However, all three have similar perceptions of the process. The process is iterative, continuing and requires control and flexibility. The development needs to follow the long-term strategy of the company, but in order to be efficient it needs to have short-term objectives. The development also requires flexibility since idea generation and programming are time consuming tasks. These models are designed for small IT companies, they are not recommended to be used as such on larger companies since the environment and resources are very different.

Partner networks was one of the research questions for this study. It was found that partner networks are used to facilitate growth and that they need to be carefully managed and monitored. Partner networks are used in software industry in particular since the industry is dominated by small companies, and they do not have the economies of scale to operate in international environment by themselves. Many small software companies have expanded rapidly to several international markets, especially if their product can be acquired through the Internet, which is the case with most of the software available.

This study focused on the description of software product development in Finnish software SMEs. More research is needed on the qualitative side to describe why certain methods work better than others. Also the commercialisation was left to minimum role, only involved to describe how it is recognised in the development process. More research is therefore needed on the commercialisation of new software products. Also the case company is not suitable to describe this topic since the focus is not on the commercialisation activities but in the customer care.

REFERENCES

Anselmo, Donald and Ledgard, Henry (2003), “Measuring Productivity in the Software Industry”, *Communications of the ACM*, 46 (11), 121-125

Antony, Jiju and Fergusson, Craig (2004), “Six Sigma in the software industry: results from a pilot study”, *Managerial Auditing Journal*, 19 (8/9), 1025-1031

Aramand, Majid (2008), “Software products and services are high tech? New product development strategy for software products and services”, *Technovation*, 28, 154-160

Börjesson, Sofia, Dahlsten, Fredrik and Williander, Mats (2006), “Innovative scanning experiences from an idea generation project at Volve Cars” ,*Technovation*, 26, 775-783

Coates, Nigel F., Cook, Iain and Robinson, Harry (1996), “Idea Generation techniques in an industrial market”, *Journal of Marketing Practise*, 3 (2), 107-118

Conway, H. Allan and McGuinness, Norman W. (1986)m “Idea Generation in Technology-Based Firms”, *Journal of Product Innovation Management*, 4, 276-291

Cooper, Robert G. (1994), “Debunking the Myths of New Product Development”, *Research Technology Management*, 37 (4), 40-50

Cusumano, Michael A. (2007), “The Changing Labyrinth of Software Pricing”, *Communications of the ACM*”, 50 (7), 19-22

Easingwood, Christopher and Harrington, Shelley (2002), “Launching and re-launching high technology products,” *Techonovation*, 22 (Fall), 657-666.

Flint, Daniel J. (2002), “Compressive new product success-to-success cycle time Deep customer value understanding and idea generation” *Industrial Marketing Management*, 31, 305-315

Gurnani, H. and Karlapalem, K. (2001), ”Optimal pricing strategies for Internet-based software dissemination”, *Journal of the Operational Research Society*, 52, 64-70

Harris, Michael, Aebischer, Kris and Klaus, Tim (2007), “The Whitewater Process: Software Development in Small IT Business”, *Communications of the ACM*, 50 (5), 89-93

Kotler, Philip and Keller, Kevin L. (2006). *Marketing Management*. Twelfth Edition. New Jersey: Prestice-Hall.

Kulmala, Harri I. And Uusi-Rauva, Erkki (2005), “Network as a business environment: experiences from software industry”, *Supply Chain Management*, 10 (3/4), 169-178

Lee, Myun W., Yun, Myung H. and Han, Sung H. (2001), “High Touch – an innovative scheme for new product development: case studies 1994–1998”, *International Journal of Industrial Ergonomics*”, 27, 271-283

Lee, Yikuan and O'Connor, Gina C. (2003), “New product launch strategy for network effects products”, *Academy of Marketing Science*, 31 (Summer), 241-245

McAdam, Rodney and McClelland, John (2002), “Sources of new product ideas and creativity practises in the UK textile industry”, *Technovation*, 22, 113-121

Rautiainen, Kristian, Lassenius, Casper & Sulonen, Reijo (2002), “4CC: A Framework for Managing Software Product Development”, *Engineering Management Journal*, 14 (2), 27-32

Ruokonen, Mika (2008), “Market orientation and product strategies in small internationalising software companies”, *Journal of High Technology Management Research*, 18, 143-156

Ruokonen, Mika, Nummela, Niina, Puumalainen, Kaisu and Saarenketo, Sami (2008), “Market orientation and internationalisation in small software firms”, *European Journal of marketing*, 42 (11/12), 1294-1315

Sands, Saul (1979), “Techniques for Creating New Product Ideas: which to choose”, *Management Decision*, 17 (2), 202-213

Sheremata, Willow A. (2002), “Finding and solving problems in software new product development”, *The Journal of Product Innovation Management*, 19, 144-158

Simpson, James T., Kollmannsberger, Christine, Schmalen, Helmut and Berkowitz, David (2002), “New product development in German and US technology firms”, *European Journal of Innovation Management*, 5 (4), 194-207

Troy, Lisa C., Szymanski, David M. and Varadarajan, P. Rajan (2001), “Generating new product ideas: An initial investigation of the role of market information and organisational characteristics”, *Journal of the Academy of Marketing Science*, 29 (1), 89-101

Zwikael, Ofer (2008), “Top management involvement in project management”, *International Journal of Managing Projects in Businesses*”, 1 (4), 498-511

Appendix 1

Interview structure in Finnish

1 Mikä on tuotekehityksen rooli yrityksessänne?

2 Mitkä ovat tuotekehityksen tärkeimmät osatekijät yrityksessänne?

3 Kuinka tärkeänä koette asiakkaan roolin tuotekehityksessä?

4 Mitkä ovat tuotekehitysprosessin tärkeimmät vaiheet yrityksessänne?

5 Mikä on tuotekehitysprosessin vaihdein suhteellinen tärkeys? Mihin vaiheeseen panostetaan eniten yrityksessänne? Miksi?

6 Mikä on tuotekehitysprosessin tavallinen kesto ideoinnista valmiiseen tuotteeseen?

7 Kuvaile tuotekehityksen valvontatoimenpiteitä yrityksessänne?

8 Kuvaile teknisen tuen merkitystä tuotekehityksessä?

9 Mikä on palautteen merkitys tuotekehityksessänne?

10 Millä tavoin ja kenelle valmis tuote markkinoidaan?

Appendix 2

Interview structure in English

1 What is the role of product development in your organisation?

2 Who are the key participants in product development in your organisation?

3 How important is the customer in product development?

4 What are the key phases in your product development process?

5 What is the relevant importance of each of these phases in your organisation? Which phase is emphasised the most? Why?

6 What is the typical length of the product development process from idea generation to the finished product?

7 Describe the management and supervision of product development in your organisation?

8 Describe the importance of tech support in product development?

9 What is the importance of customer feedback in product development?

10 To whom and how is the finished product promoted?

Appendix 3

Haastattelu @ Yritys X

Haastattelija: Mikä on tuotekehityksen rooli teidän yrityksessänne?

Vastaaja: Elikkä tota, eli, meidän yrityshän on ohjelmistotuotetoimittaja. Elikkä meille tuotekehitys on ollut jo vuodesta 1991 lähtien erittäin suuressa merkityksessä. Sillä on ollut erittäin suuri merkitys. Tällä hetkellähän tuotekehityspanostukset on noin 21% liikevaihdosta. Suurin osa meidän rahoistahan tulee lisenssimyynneistä ja sitte täst maintenanssista, joka siis on näiden uusien päivitysversioiden toimittamista asiakkaille. Elikkä tää ihan perusohjelmistotuotteeseen liittyvät tuotekehitys on erittäin keskeisessä roolissa.

H: Mihin tuotekehitysprosessi pyrkii?

V: No siis meil on täl hetkellä tilanne sellanen et meil on käytännössä kaksi erillistä ohjelmistotuotetta, jotka sitten käyttää kohtuullisen paljon yhteisiä komponentteja, mutta kaks ohjelmistotuotetta on se lopputulos ja niistä tehdään eri leaseja, suurinpiirtein kerran vuodessa tämmönen isompi release ja sitten tehdään tietysti pienempiä päivitysversioita. Mutta kaikki mitä tuotekehitys tekee niin on siihen samaan branchiin, elikkä me ei tehdä mitään asiakaskohtaisia omia tuoteversioita, vaan et kaikki ominaisuudet mitä tehdään tuotteeseen tulee aina samantien kaikille asiakkaille. Kaikki ei tietenkään kaikkia käytä.

H: Millä tavalla tuotekehitys on huomioitu yrityksen strategiassa?

V: Tuotteet on kokolailla geneerisiä, eli niitä voi käyttää hyvinkin moneen eri tarpeeseen. Strategianäkökulmasta se panostuksen määrä mitä tuotekehitykseen laitetaan, ni se nyt on ollut suhteellisen tasaista. Meil on tuotekehitysorganisaatio ja siel on tietyt panostukset ja et se on säilyny kohtuullisen tasasena, mut sitte se mitä strategiassa koko ajan todella paljon mietitään ni on se että mihin asiakastarpeeseen me halutaan suunnata tätä businessta. Se tarkottaa sillon sitä että sillä on välitöntä vaikutusta siihen et miten me markkinoidaan ja sit siihen et minkälaisia partnereita me haetaan, mihin me meidän myyntieffortit laitetaan, mut sit se on myöskin olennainen osa sitä pitkäntähtäimen planningia missä määrätään minkälaisia tuoteversioita me ruvetaan tekemään, mitä

ominaisuuksia sinne tuotteeseen tulee. Eli se on niinku se strategiakytkentä eli se mihin suuntaan strategiassa ollaan menossa ni sehän on sitten se mihin pidemmälläki tähtäimellä uusia ominaisuuksia ruvetaan tekemään. Mut lähinnä siis strategia määrää sen suunnan että minkälaisia asioita tuotekehityksessä jatkossa tehdään.

H: Minkälaisia rajotuksia ja mahdollisuuksia partneriverkon käyttäminen ulkomailla tuo tuotekehitykseen?

V: Eli 2/3 liikevaihdosta tulee ulkomailta. Rajotukset on semmosia et kun meillä on lukumääräsesti paljon partnereita ketkä sitten tekee toisaalta kohtuullisen vähän kauppoja per partneri, eli partnerit tarjoaa siihen sen konsultointikerroksen ja palvelut siihen päälle ja saavat siitä ison osan rahaa, se softamyynti ei välttämättä oo se mikä sen partnerin kokonaisuudesssaan elättää. Mistä taas aiheutuu se et partnereiden kompetenssit sen varsinaisen tuotteen käyttämiseen on niinku rajalliset, eli hei ei kovin suuria määriä pysty pistä aikaa sen tuotteen opetteluun ja tää taas tarkottaa et tuotteen pitää olla helppokäyttönen ja tää on meillä ollu alusta asti näitten omien tuotteiden osalta et tuotteiden pitää olla tosi helppokäyttösiä ja helppokäyttöset tuottee, ni asiakkaat tykkää niistä ja niitä on myös mahdollista myydä tämmösen partnerikanavan kautta. Välillä siinä joudutaan tekemään kaikenlaisia API-rajapintoja ja muita webservice tyyppisiä vaikeita asioita, mut nää on yleensä semmosia mitä partnerit ei pysty kovin hyvin hyödyntämään. Sit taas toisaalta mikä on relevanttia, ni tää softa on käännetty kahdellekymmenelle kielelle ja täs on tää multilizer-tyyppinen käännös, eli nykysin on helppoa kääntää käytännössä mille tahansa kielelle. Ja tähän osallistuu partnerit tähän lokalisaatio-prosessiin. Et he on itse osa sitä lokalisointia.

H: Ketkä osallistuu tuotekehitysprosessiin teidän yrityksessänne ja ketkä ovat tärkeimmät osallistujat?

V: No, product management on meillä se yksikkö joka vastuussa siitä minkälaisia uusia ominaisuuksia meillä tulee ja siellä tehdään road mappausta aina käytännössä kaikille uusille isoille versioille, mitkä tarkottaa et yhdeksän kuukauden tai kahdentoista kuukauden päässä tulevalle versiolle. Tehdään isompaa road mappausta siitä et mitä sinne laitetaan. Road map tehdään ehkä kolmen vuoden päähän. Mut tää on sen product managementin tehtävä olla suodattamassa sitä informaatiota, mitä tulee asiakkaalta, partnereilta ja yrityksen johtoryhmästä ja strategioiden kautta. Kaikkien näiden kautta

product management toimii sulatusuunina ja tekee päätöksen siitä mitä toteutetaan. Sitten kun product management on saanu tehtyä päätöksen siitä mitä toteutetaan niin sitten on product development yksiöitten vuoro ja siellä meillä on muutama development tiimi, laadunhallinta tiimi ja production tiimi jossa sitten tää tuotekehitys tapahtuu. Siellä on 20-24 ihmistä jotka tekee tätä tuotekehitystä.

H: Kummalla on suurempi rooli tuotekehityksessä: asiakkaalla vai yrityksen sisäisillä toimijoilla, teknisellä osaamisella?

V: Jos tuolla lailla kysymys asetellaan niin asiakas. Ehkä voisi ajatella asiaa siten että me ollaan enemmän markkinaohjautuva kuin asiakasohjautuva. Jos tuotekehitystä voisi ohjata markkinat, asiakas tai teknologia, ni me ollaan markkinaohjautuva. Me tehdään niitä ominaisuuksia mitä me uskotaan että markkinoilla kaivataan, me ei tehdä yksittäisen asiakkaan vaatimuksia niin paljoa. Me saadaan niin paljo toivomuksia ettei me pystytä niitä kaikkia tekemään. Joudutaan kattomaan sitä kokonaisuutta.Sit se mitä teknologia mahdollistaa niin se on pienemmässä roolissa kuin mitä asiakkaat haluaa. Toki jos teknologinen toteutus on liian vaikeaa ni sitä ei lähdetä tekemään.

H: Mitkä on tuotekehitysprosessin tärkeimmät vaiheet?

V: Ensin on mahdollisten uusien tuoteominaisuuksien tunnistaminen ja löytäminen ja siihen on monia tahoja josta tietoa suodatetaan. On asiakkaita, partnereita, yrityksen johto ja markkinoiltakin katotaan että mitä kilpailijat on tehneet. Käytännössä homma etenee niin että product management luo sinne sitten uudet ominaisuudet vaatimustenhallintatietokantaan ja sitten ensimmäinen on toiminnallisuuden speksaaminen käyttäjän näkökulmasta ja sitten teknisestä näkökulmasta. Teknisessä speksaamisessa ominaisuudelle saadaan työmääräarvio. Tässä vaiheessa product management tekee päätöksen että otetaanko ominaisuus mukaan seuraavaan releaseen vai ei. Kyseessä on siis aikatauluttaminen, eli jos ei seuraava release ni sitten sitä seuraava, kovin pitkälle ei kiinnitetä tulevia versioita. Kun ominaisuus on hyväksytty releaseen sille tehdään tekninen suunnittelu. Sen jälkeen kun päätös on tehty, kokenut developer katsoo mitä tää vaatii ja mitä tulee ottaa huomioon kun tämä toteutetaan. Sitten tulee varsinainen koodaamistyö. Ensin koodataan ja sitten tulee code review, minkä tekee joku toinen developer. Sitten ruvetaan tekemään testausta. Eli koodaus, code review ja testaus on yleensä eri henkilöitten suorittamat. Voi olla niin simppeleitä ominaisuuksia jotka on niin

itsestäänselviä että menee helpommallakin. Testien läpimentyä tulee tuotteistamiseen liittyvät asiat, eli tehdään linkit sinne dokumentaatioon ja sitte tulee tähän lokalisaatioon liittyvät asiat ja lopulta sitten ollaan käytännössä valmiita ottamaan se [ominaisuus] mukaan siihen releaseen. Eli tässä nyt yksittäisen featuren näkökulmasta katottuna toi tuotekehitysprosessit.

H: Onko teillä asiakastestiryhmää?

V: Yllämainittu testaus on yksittäiselle featurelle. Sitten ton jälkeen kun se feature on testattu ni tulee field testaus product managementille eli kun joku tietty ominaisuus toimii ni katotaan että toteuttaako se sen tarpeen mitä varten se alunperin speksattiin. Ni tässä tulee change reviewta ja change requestia. Varsinainen ulkopuolinen testaus alkaa siitä että kaksi tai kolme kuukautta ennen varsinaista releasea me lähetetään ensimmäinen versio partnereille lokalisointia varten. Eli partnerit on meillä tämmönen taho joka myös testaa tuotteita ja partnerit testaa asiakkaiden malleilla niitä tuotteita, eli partnereilla on asiakkaiden tietokantoja joilla ne kattoo että toimiikohan tää sovellus vielä. Sama juttu ku mitä Suomessa tehdään, ni sillä release kandidaatilla katotaan että mitenköhän tää toimii näitten meidän suomalaisten asiakkaiden kanssa. Tän perusteella saadaan tehtyä se lopullinen release. Mut loppuasiakkaat ei ota testiympäristöä jossa he testais, eli testauksen tekee meidän yritys, tuotekehityksen alihankkijat ja partnerit.

H: Ovatko tuotekehitysprosessin vaiheet samanarvoisia vai panostetaanko johonkin vaiheeseen enemmän?

V: Meillä on aika hyvä statistiikka siitä et kuinka paljon kuhunkin featureen on käytetty aikaa. Ehkä peli on siitä kiinni et kuinka fiksusti product managementissa osataan tehdä päätöksiä siitä et mitä tänne ylipäätänsä tehdään ja mitä jätetään pois ja et saadaan semmosia hyviä featuresettejä mistä on käytännön hyötyä todellisen asiakastarpeen ratkasemiseen. Muutoin jos ajatellaan viiden vuoden aikana tapahtuneita muutoksia ni testaukseen on panostettu, mutta sekin näkyy sillälailla et meillä on nyt pari kolme vuotta ollu hyvässä kunnossa noi automaattiset testausympäristöt, meillä uus tuotekehityksen build tehdään kolme kertaa päivässä ja sitten ne buildit menee aina tonne automaattiseen testaukseen et kyl siellä aikalailla isoa regressiotestausta pyöritetään koko ajan, suurinpiirtein vuorokauden ympäri. Mut toki tehokkuutta haetaan jokaisesta vaiheesta, tehokkuutta haetaan muista vaiheista ja sit product management on kiinni siitä älyllisestä

toiminnasta.

H: Taisitte mainitakin jo, eli teillä on releaset 9-12 kuukauden välein?

V: Joo.

H: Mutta entäs sitten ihan yksittäisen idean työstäminen, kuinka pitkä kestoinen se tavallisesti on?

V: No, perussääntö on se että uusia ominaisuuksia tulee vaan niihin uusiin releaseihin. Eli perussääntö on et siin matkal ei tuu uusia perusominaisuuksia. Toki niitä uusia ominaisuuksia tehdään, mut rajotteita on se että me ei haluta tehdä mitään muutoksia meidän tietokantaan tän 9-12 kuukauden jakson sisällä, eli tää uuden version käyttöönoton tän jakson välillä täytyy olla asiakkaille tosi yksinkertanen operaatio eli ajetaan vaan tämmönen service pack sisään. Niissä voi joskus olla jotain pieniä ominaisuuksia lisättynä tai muutettu jotain toiminnallisuutta, mutta sitten varsinaisesti jos puhutaan uudesta toiminnallisuudesta ni tosiaan tulee näihin isompiin versioihin. Niis on semmonen sykli et ne on holdissa siihen asti kunnes se road mappi alkaa, eli katotaan mitä oikeesti aletaan tekemään siihen seuraavaan versioon. Se tarkottaa keskimäärin sitä että siitä kun joku ominaisuus otetaan mietintään ni menee tyypillisesti se 9-12 kuukautta ni meil on se ominaisuus käytettävissä. Me ei tehdä kovin paljoa tämmöstä jollekin tietylle asiakkalle häthätää juuri halutaan saada joku ominaisuus versioon, tämmöstä työtä me pyritään tekemään kohtuullisen vähän. Tällä pyritään siihen hyvään tehokkuuteen, et saadaan tehtyä niit isoja kokonaisuuksia ja tuotetta määrätietoisesti kehittyy siihen suuntaan mihin me nähdään et sitä kuuluu kehittää. Ettei aleta riskeerata tätä määrätietosta kehittämistä sillä et jotkut pienet urgentit asiakasvaatimukset tunkis siihen edelle. Et jos haluaa tehdä jotain asiakaskohtasia räätälöintejä ni niitä voi tehdä sitten niitten ohjelmistorajapintojen kautta, jos on niin kova tarve et pitää heti saada jotain toimivaa asiakkalle.

H: Millä tavalla tuotekehitystä valvotaan, minkälaisia toimenpiteitä siihen on?

V: No meil on aika hyvä tuotehallintatietokanta, joka ulottuu koko tuotefeatureiden tekemisen elinkaaren ajalle, eli tavallaan kaikki noi vaiheet tallettuu tietokantaan ja nähdään et kuinka paljon kukakin on siihen tehny. Tosta saadaan aika hyvä mittaridata ja se on myös bonuskäytännöissä mukana, ei kauheen isolla painolla, mut onpahan kuitenki.

Kyllä ne luvut laskettua tulee kuitenkin, tehokkuusmielessä. Laatua seurataan sillä et paljonko niitä bugeja löytyy sieltä valmiista tuoteversiosta. Bugeilla on oma luokituksensa ja myös niitä seurataan ja niihin on asetettu tavotteita. Se mitä on vaikee seurata ja johtaa on se et mitä ominaisuuksia sinne on valittu ja mitä tehty. Se on absoluuttisesti haasteellinen asia. Siitä ei ehkä objektiivisesti voi sanoa kun että me tehdään asiakastyytyväisyyskysely kerran vuodessa, se kertoo siitä miten me ollaan onnistuttu vastaamaan asiakkaiden tarpeisiin. Sitten meillä asiakkaat keskimäärin, maksaa 20% lisenssihinnasta on tän vuotusen maintenancen suuruus. Sit tietysti se et kuinka pitkään asiakkaat keskimäärin pitää maintenancea voimassa ni kertoo myös siitä kuinka hyvin ne on nähny arvoa sille maintenancelle, eli uusille versioille. Eli noita kahta asiaa seurataan, enemmän ehkä tyytyväisyyttä ja tähän maintenancen keskimäärästä pituutta seurataaks eniten. Sitte meil on tietokannas tieto siitä miten asiakkaat on ottanu uusia versioita käyttöön, ni me nähdään et kuinka moni on vielä vanhoissa versioissa ja sehän kertoo myös siitä että ollaanko onnistuttu tekemään juuri sellasia ominaisuuksia mitä ne asiakkaat niin kipeästi tarvii. Noi mittarit on vähän haasteellisia, esim benchmarkki-dataa on vaikea saada, on vaikea verrata meidän ja jonku toisen tuotteita. Paremmin me saadaan tuotteita markkinoille ku esimerkiks Microsoft on viime aikoina saanu omia uusia käyttöjärjestelmiä markkinoille, eli kyl meillä keskimäärin asiakkaat ottaa tyytyväisinä uudet versiot käyttöön.

H: Jos verrataan valvontaa hierarkkisesti, niin kuinka paljon ylimmällä tasolla otetaan kantaa, kuinka paljon siinä välillä ja kuinka paljon valvontaa tapahtuu ihan lähiesimiestasolla?

V: Kyl se valvonnan tavote on enemmän se hyvä laatu ja sitten sen prosessin tehokkuus tulee niinku kakkosena. Ne mittarit tekee valvonnasta kohtuu helppoa. Kyl meil developerit on aika itsenäises asemas, ei siellä tavallaan tarvita ketään siihen viereen kyttäämään et teetkös nyt työtäsi oikein tai teetkös työtäsi dokumentaatioitten mukaan. Esim se code reviewkin on vaan nähty sellasena et se on yksi hyvä tapa nähdä mahdollisia bugeja ja myös kompetenssia kehittää tiimin sisällä, sillä et kun yks tiimin sisällä on koodannu ni toinen kattoo et mites se on sen koodannu ja mitäs tästä vois muuta. Mä ajattelisin että aika itsenäistä tuo työ tuolla on, kaikki tulokset on loppujen lopuksi siinä tuotteessa. Kyllä sen sitten kuulee jos siellä on hölmösti speksattuja featureita tai jos siel on bugeja tai jos dokumentaatio ei oo ajantasalle ni kyllä meidän asiakaskunnasta tuppaa se tieto aikanaan tulemaan.

H: Minkälainen merkitys teknisellä tuella on tuotekehitysprosessissa? Onko se edes osa tuotekehitystä?

V: Bugeja on valmissoftassa mittavat määrät, niitä luokitellaan ja korjataan. Me tehdään service pack versioita noin puolentoistakuukauden välein, jos ajatellaan että meil on se yhdeksän kuukauden sykli, ni siit tulee noin kuus kappaletta service packeja tähän releaseen ennenku tulee se isompi versio. Ja sitten vielä ku se uusi versio tulee, ni me tehdään vielä näitä service packeja vanhaan versioon. Eli kyl tuol aika paljon tehdään työtä myös vanhoihin tuoteversioihin liittyviin bugikorjauksiin. Et siel on kohtalaisen monta code basea auki mihin sitä työtä tehdään. Ja kylhän sen maintenancen ja supportin iso osa on toki se että nää kaikki korjaukset sieltä sitten tulee. Kyllähän se aikaa selkeesti vie ja mitä enemmän täs on kokoajan tullu asiakkaita ni sen enemmin ja paremmin sieltä pienetki bugit löytyy ja pienetki bugit joitain asiakkaita häiritsee ja mielellään pienetki bugit korjataan. Se vie tietyn vakiotyyppisen määrän noista tuotekehityksen resursseista koko ajan. Me ollaan aika paljon teetetty ulkopuolisella hankkijalla näitä bugikorjaustyötä, se on kohtuullisen suoraviivasta työtä ja helppoa valvoa. Väärinymmärrysten määrä on suhteellisen pieni kun voi näyttää että tämä softa toimii tässä tilanteessa näin kun sen pitäis toimia noin, ni se on kohtuullisen helposti kommunikoitavissa tää äskeinen. Mut kyl se selkeesti tuotekehitykseen kuuluu. Sinänsä, maintenance&supportiin kuuluu myös tää customer care tuki, joka ei ole meillä tuotekehitystukea vaan asiakkaiden auttamista käyttämään sitä softaa. Mutta kuten todettu, peruskäyttö on aika yksinkertasta, siitä tulee vähemmän kysymyksiä ja sitte yleensä nää customer care requestit liittyy johonki meidän tuotteen yhteiskäyttöön joittenki muitten tuotteitten kanssa ja sillon pitää tuntea niitä muita tuotteita ja se on vähän kauempana meidän omasta tuotekehityksestä.

H: Tuleeko teillä teknisen tuen kautta uusia kehitysideoita vai onko se enemmän virheiden korjausta?

V: Customer care myös yleinen sähköpostiosote mihin asiakkaat ja partnerit lähettää ajatuksia, erityisesti partnereilta tulee toiveita, jotka on lähtösin alunperin asiakkaiden toiveista ja suurimmat asiakkaatki lähettää suoraan. Se että onko se nimenomaan customer care vai joku meidän oma konsultti joka on ollut sitä asiakasta auttamassa, ni tyypillisimmin se on se oma konsultti joka on auttanu. Ehkä voidaan ajatella että semmonen muutos on tapahtunu et jos jos kymmen vuotta sitte asiakkaat oli viel niinku erityisen innokkaita auttamaan meitä ja kertomaan meille suuren määrän ideoita mitä me

voitas tehdä, ehkä nyt on asiakkaillaki kiire painaa päälle ja ylimäärästä aikaa on vähemmän ja asiakkaat on ehkä suhteessa vähemmän valmiita sitä aikaa käyttämään, ni sillo kun oma konsultti käy asiakkaan luona juttelemassa ni löytyy enemmän asioita joista me pystytään ite poimimaan ne tuotekehityksen uudet ajatukset. Meillä on tosiaan tämmönen customer care tiimi joka toimii semmosena frontline helpdesk palvelua tarjoavana tiiminä, mutta maintenancea ja supportia antaa myös konsultit, myyjät ja partnerit. Sitä kautta tulee iso osa meidän tuotekehityksen toiveista.

H: Kuinka tärkeä osa asiakaspalaute on tuotekehitystä?

V: Joskus laskettiin että 80% ominaisuuksista mitä tehtiin oli sellasia mitä tehtiin asiakkaiden toivomuksesta ja 20% oli semmosia mitä ei tehty erityisesti asiakkaiden toivomuksesta. Asiakaspalaute on erityisen tärkeätä. Asiakkaat toivoo tosi paljon kaikenlaisia asioita ja siinä on se oman vision muodostaminen et missä tän tuotteen pitää olla kahden kolmen vuoden kuluttua, ni se on se keskeisin asia. Asiakaspalaute muutoin on ensiarvoisin tärkeätä, esimerkiks täst käytettävyydestä ja helppokäyttösyydestä. Siinä on tosi tärkeetä mahdollisimman hyvin palautetta asiakkailta et nyt ne on ottanu ton ja ton ominaisuuden käyttöön tähän ja tähän tarkotukseen et mitäs ne on saanu siel aikaseks ja mitä tuumaillu. Ne on erittäin tärkeitä tietoja.

H: Kenelle service packeja markkinoidaan ja kenelle releaseja ja millä tavalla katetaan tuotekehityksen kulut?

V: Service packien ja releasejen hinnoittelu on meillä yleensä juuri täst kanavaluonteesta johtuen pyritty pitämään mahdollisimman yksinkertasina. Eli, service packit, niitä ei kenellekään koskaan erikseen myydä vaan ne aina sisältyy siihen maintenance & support serviceen. Maintenance ja support on tehty niin että ensimmäiseks vuodeks on pakollista ostaa se maintenance ja support. Jokaisella jolla on meidän softatuote käytössä ni tyypillisesti on maintenance ja support käytössä, ei se pakollista ole. Service packit asiakas ottaa käyttöön sitä mukaan kun he ohjelmistossa törmää johonki bugiin tai käytettävyysongelmaan jonka se service pack korjaa. Tää on tyypillisin syy ottaa service pack käyttöön. Jotkut eturintaman asiakkaat ottaa niitä uusia service packeja ihan niiden pienten parannusten takia mitä siellä on tehty ylipäätänsä, koska se service packin käyttöönotto on niin helppoa. Tai sitte jos on aktiivinen partneri ni partneri ikäänku saa sen asiakkaan haluamaan niitä ominaisuuksia ja pystyy hyödyntämään niitä ominaisuuksia

siinä konsultoinnissa. Muuten niitä service packeja ei erikseen koskaan myydä et niille ei lasketa erikseen mitään arvoa. Yleensä se service pack julkistetaan siinä vaiheessa ku meidän bugiluokituksen mukaan ohjelmistosta on löytyny joku critical bugi, jonka seurauksena halutaan bugikorjaus jolleki asiakkaalle toimittaa ja sitä varten tehdään ihan virallinen service pack. Uudet releaset, ni niitten kanssa partnerit ja loppuasiakkaat voi tehdä jotain kustomointeja uusiks mitä he on tehny tähän asti. Siinä puhutaan helposti yhen kahen kolmen, isoimmilla asiakkailla kymmenen, parikytä päivää jos on tehty paljon asioita ja halutaan kaikki uudet asiat omaan käyttömenetelmään huomioida, että miten tätä ominaisuutta nyt meidän ympäristössä käytetään. Keskimäärin se on jonkilainen projekti ni asiakas joutuu tekemään business päätöksen että siirrytäänkö tähän uuteen versioon vai mitä sen suhteen tehdään. Monet asiakkaat saattaa tehdä testiympäristön erikseen missä testataan se toiminta että se omaan toimintaympäristöön sopii se tuote. Me ei kyllä lasketa erikseen niille mitään erillista rahallista arvoa niille releaseille tai pidetä mitään kirjaa että tätä ja tätä releasea on myyty näin ja näin paljo. Enemmän nähdään se semmosena jatkumona. Tuodaan kyllä esille markkinoinnissa uus tuote, mutta kun asiakas ostaa niin tilauskaavakkeissa ei lue sitä versionumeroa minkä he ostaa. Se versio minkä he ostaa on mikä sillä hetkellä on voimassa oleva tai mikä tahansa aiempi versio tai sitte mikä tahansa sen vuoden aikana tuleva ylläpito versio. Ja kun ylläpito jatkuu ni voi käyttää haluamaansa versiota.

Appendix 4

Interview @ Company X

Interviewer: What is the role of product development in your company?

Respondent: Our company is a software product supplier, so for us product development has been, since 1991, very important. It has had a very significant impact. Currently investments into product development are about 21% of our turnover. Most of our revenue comes from licence sales and maintenance, which consists of delivering new upgrades to customers. So product development concerning the basic software product is in a very important role.

I: What is the goal of product development?

R: At the moment we have two software products. They use relatively many common components, but the two software products are the end results. Separate releases are made of them, about once a year a major release and then of course smaller upgrades. But everything that the product development does is for the same branch, we don't do any customer specific releases, but all the features that a certain product will have will be available for all the customers. Naturally not all are used by everyone.

I: How is product development taken into account in the strategy of your company?

R: The products are rather generic, so they can be used in many ways. From the strategy point of view the investments put into product development have been rather steady. We have a product development organisation and their contribution has been rather steady, but what is constantly an issue for the strategy is to which customer need we wish to strive this business for. It means that [strategy] has an immediate impact on how we market [our products] and what kind of partners we are looking for and where do we put our sales efforts, but it also has a significant part on the long-term planning of product versions, what kind of features are included. So that's the connection with the strategy; where we are going with the strategic decision means what new features are going to be included in the products. But mostly our strategy defines the direction of what we do in the product development in the future.

I: What kind of limitations and opportunities your international partner network brings to product development?

R: So 2/3 of our turnover comes from abroad. Since we have a large number of partners who do relatively little sales per partner, so they provide the consulting layer and services and generate substantial sales through those, Therefore the sales of the actual software product are not very important to the partner. This means that the competences of our partners to use the software are limited, do they are not able to spend a lot of time learning to use the software so they software needs to be easy to use. This has been our aim from the beginning for our products, they need to be easy to use and the customers also appreciate those and they are easy to sell through this partner network. Occasionally we have to all sorts of API-interfaces and other webservice type of more difficult things, but these are usually something that our partners can't take advantage of. Another relevant point is that our software is translated into twenty languages and we are using this MultiLizer translation so nowadays it's easy to translate into any language. And our partners are involved in this localization process.

I: Who are the participants of the product development process and which are the key participants?

R: Well, product management is the unit who is responsible for what new features are introduced and they make road mapping for all the new versions, meaning the next version in nine or twelve months. The road mapping consists of plans on new features added and it's made for about three years. But the purpose of product management is to filtrate the information that comes from the customers, the partners, the top management and the strategy. The product management through all of this works as a melting pot and makes the decision on what to do. Once the product management has decided what to do, then the product development units, who consist of development team, quality control team and production ream, conduct the actual development. There are about 20 to 24 people who do this development.

I: Which has a bigger role in the product development: the customer or the internal actors, technical knowhow?

R: If the question is stated like that then the customer. Maybe you could say that we are

more market-oriented than customer-oriented. If product development could be market-, customer- or technology-oriented, then we are market-oriented. We develop those features that we think that the market needs, we do not include individual customer requests so much. We get so many requests that we cannot do them all. We have to look at the wholeness. What technology can offer is of minor importance than what customers want. For sure if it's difficult to do then we don't do it.

I: What are the key phases in product development?

R: First is the identification and finding of potential new product features and there are many sources for this information. [it can be] the customers, the partners, the top management and we also look into the markets to see what our competitors have done. What actually happens is that the product management creates the new features into the Bugs&Feature database and then what follows first is the operational specification from the user point of view and then from the technical point of view. In the technical specification a feature receives a workload estimation. At this point the product management decides whether to include the feature in the next release or not. This is called timing, so if it's not included in the next relase, then perhaps in the one after that. We don't plan too far ahead. Once the feature is accepted into the release a technical planning is made. After the decision is made an experienced developer estimates what needs to be done and what needs to be considered when the feature is included. This is followed by the actual coding. At first the feature is coded and then the code is reviewed by another developer. Then the code is tested. So the coding, code review and testing are usually done by different people. There can be so simple features that are so obvious that things can be done easier. After the tests are ok, actions related to turning the code into tangible product follow. These include creating links to documentation and localization actions and finally we are ready to include [the feature] in the release. So this is the development process from the view point of a single feature.

I: Do you have a customer test-groups?

R: The above testing is for single features. The feature testing is followed by a field test done by product management, meaning that after we have discovered that certain feature is working we need to find out whether it solves the need it was originally specified for. This causes change reviews and change requests. The actual external testing begins

about two or three months prior to the actual release when we send the first version to our partners for localization purposes. So partners are also some form of testers, since they are testing the models of our customers on the new release, since they have some databases of our customers and they use them to test whether the new release still works. It's the same thing here in Finland, we test the new release candidate with our Finnish customers to see how it works. Based on this we are able to produce the final release. But our end customers don't test our releases, the testing is done by our company, our product development sub-contractors and partners.

I: Are the phases in product development process equal or do you place more emphasis on a particular phase?

R: We have rather good statistics on how much time is spent on each feature. Perhaps the key issue is how smart decisions product mangement is able to make on what to include and what to exclude so that we get good feature sets which are beneficial in solving a real customer need. Otherwise if we consider the changes during the past five years, the testing is something that we have put much effort into. That is shown best in our well-performing automatic testing environment, we've had that for the past few years now. We make a new product development build three times a day and each build goes into the automatic testing so we are doing rather large regression analysis all the time, around the clock I would say. But for sure we are searching efficiency from all the phases, efficiency from all the other phases and product management is dependent on the intellectual actions.

I: You already mentioned, but you have releases every nine to twelve months?

R: Yes.

I: But what about a single idea, how long it usually takes to process that?

R: Well, the principal is that new features are only included in the new releases, so we do not include any new basic features in between releases. Of course we make new features, but they are limited by our desire not to make any changes into our databases during this nine to twelve months, because we want the implementation of this new version to be as easy as possible for our customers, all they need to do is run a service pack. Some minor

new features might be added or some functionality might be altered, but if we are talking about a new functionality, it will be in the next version. The new features are on hold until the road mapping begins, when we decide which features will be included in that version. This means that when a certain feature is taken into consideration, typically nine to twelve months later it will be available to our customers. We try to minimize work based on a single customer request, because we strive for good efficiency, to be able to make large entities and to develop the product purposefully in the direction we feel that it needs to be developed. We don't want to risk this purposeful development by responding to minor customer requests. If someone wants to do some tailoring for some customers it can be done via the software interfaces, if the need is so urgent that something needs to be done immediately.

I: How do you monitor product development? What kind of measures you have?

R: Well, we have a rather good product management database which covers the whole life cycle of producing new product features, so all the phases are recorded in the database and [from there] we can see who has done what. It's a really good measurement and we also use it in our bonus scheme, even though only with minor input. We calculate the numbers anyhow, efficiency in mind. The quality is monitored by the number of bugs in the final product version. The bugs have their own classifications and they are monitored and separate targets are set for them. What is difficult to follow is what features are chosen and what are left out. That's absolutely the most challenging task. What can be said objectively is that we conduct a customer satisfaction survey once a year, this tells us how we have managed to respond to customer needs. Then our customers pay about 20% of the licence price as maintenance. The length of the maintenance contract tells us how much value customers place on maintenance and therefore to new versions. So those two things are monitored, more about the satisfaction and then the length of maintenance. We also have information in our database about which customers have included the new versions [in their systems] so we can see who are still using the older versions and that tells us whether we have managed to create such features that our customers need. These measures are rather challenging since, for example benchmarking data is very hard to come up with, it's very hard to compare our products to some other company's products. We have been better in creating new products to the markets than Microsoft has been able to sell its latest system lately, so our customers are rather satisfied with our new versions.

I: If we compare the monitoring hierarchically, how much input does the top management have, how much the middle level and how much monitoring is done at the supervisor level?

R: The main purpose of monitoring is good quality and the efficiency of the process comes second. Our measurements make monitoring rather easy. Our developers are rather independent, there's no need for someone to watch them over. For example the code review is something that we have seen as a good way to locate potential bugs and to develop the competences within a team, by assigning one to do the coding, then someone else reviews how it's been made and what else could be done. I would say that the work out there is very independent, all the results are in the final product. We will find out if some features' specifications are stupid or if there are bugs or if the documentation is not up-to-date; our customers will inform us eventually.

I: What kind of impact does the tech support have on product development? Is it even a part of the process?

R: There are lots of bugs in ready-made software, they are classified and fixed. We introduce service pack versions every month and a half, so in the case we have the nine month cycle for releases, we have about six service pack versions for each release. And even after the new version, we still make service packs for the older versions. So they do quite a lot work on older versions as well, trying to fix the bugs. They have quite a many open code bases that they are working on. A very big part of maintenance and support is fixing all these problems. It takes a lot of time and the more we have customers the better we have found even the smallest bugs and even the smallest bugs irritate some customers and we gladly fix them. It takes a certain fixed amount of product development resources all the time. We have outsourced quite much of this bug fixing work, since it's relatively streamlined work and easy to monitor. The number of misunderstandings is relatively small when you can demonstrate that this software functions like this when it's supposed to function like that, so it's very easily communicated. But it's clearly a part of product development. Maintenance and support also include customer care support, which is not a part of product development in our company but helping customers using our software. But as said, the basic usage is rather simple, we don't get many questions concerning that. Usually our customer care requests are concerned with using our software with some

other software and in that case you need to know a little about the other software and that is a further away from our own product development.

I: Do you get new development ideas through tech support or is it more concerned with fixing bugs?

R: Customer care is also our general email-address where customers and partners can send ideas, and especially partners send wishes, which originate from customer wishes and our biggest clients also send direct suggestions. Whether [new ideas came from] the customer care or our own consultant who has been helping our customers, it's usually our own consultant. Maybe we could say that when ten years ago our customers were very eager to help us and tell us a large quantity of ideas on what we should fo, maybe nowadays the customers are too busy and they have less extra time than back then. So when our own consultant visits them and talks with them we are able to find the ideas that we can do. We have a customer care team who acts as a frontline helpdesk, but maintenance and support is also provided by consultants, sales staff and partners. Through them we get a big part of our product development wishes.

I: How important is customer feedback in product development?

R: We once counted that 80% of new features made were such that were made based on customer requests and 20% were such that were not made specifically based on customer requests. Customer feedback is very important. Customers are requesting many different things so we need to form our own vision of where we wish to be in few years with this product. Customer feedback is important in other ways as well, for example concerning the feasibility [of the product]. It's very important to get feedback from the customers about their level of usage of different features and to what purpose and what have they accomplished. This is very important information.

I: To whom are the service packs marketed and to whom the releases and how do you cover the costs of product development?

R: We have tried to keep the pricing of service packs and releases very simple because of our channel network. We never sell the service packs to anyone separately, but they are

included in the maintenance and support service. This is done so that the first year is mandatory [when you purchase our software]. Usually everyone who has our software also has our maintenance and support service, yet it's not mandatory. The customers include the service packs in their systems when they encounter a bug or feasibility problem that the service pack fixes. This is the most common reason for including our service packs. Some front-end customers include the new service packs just because of the small improvements that are made, because the inclusion of new service packs has been made so easy. Also if the partner is very active they can make the customer want those features and they can use these features in their consulting. Since the service packs are never sold, we don't allocate any value to them. The service pack is usually released when we encounter a critical bug in the program and we wish to provide a resolution for a particular customer so we make an official service pack for that. With the new releases our partners and end customers might have to redo some of the customizations. This can easily take few days, even ten to twenty days with bigger customers, if they have customised a lot and want to customise all the new features. Since it's a project of some effort, the customer needs to make a business decision whether to change into the new version or not. Some customers create a separate testing environment to see whether the new versions fits their environment or not. We don't allocate any monetary value for the releases, or keep any record of the sales of the different versions. We see it more as a continuity. We will include the new product in our marketing, but when our customers order our products, they don't specify in the form that which version they are ordering. The version can be the one currently in the markets or any previous version or any of the upcoming versions. And since the maintenance continues, they can use whichever version they like.

Appendix 5

Reference division by year of publication and type of publication

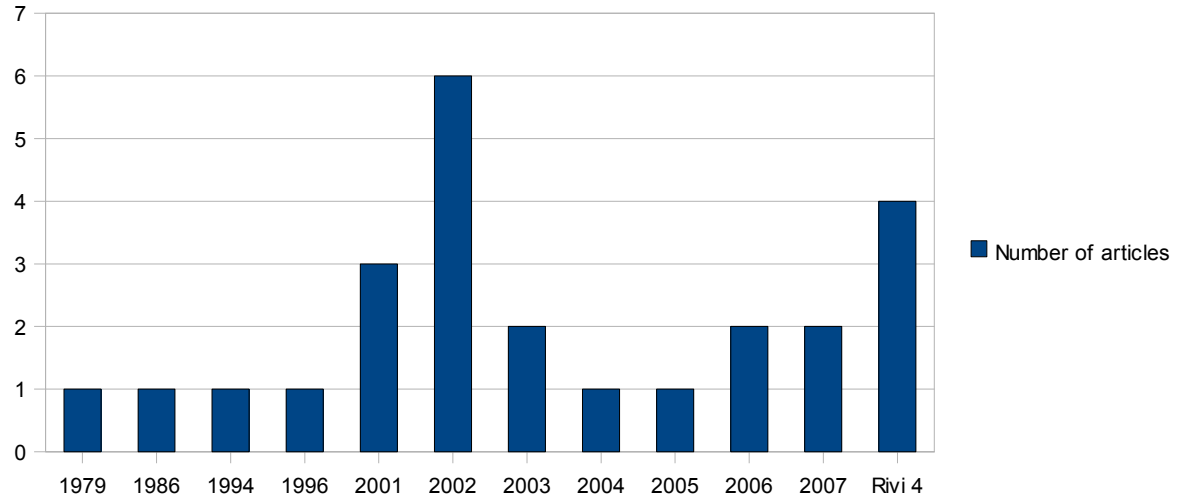


Figure 6. Articles by year of publication

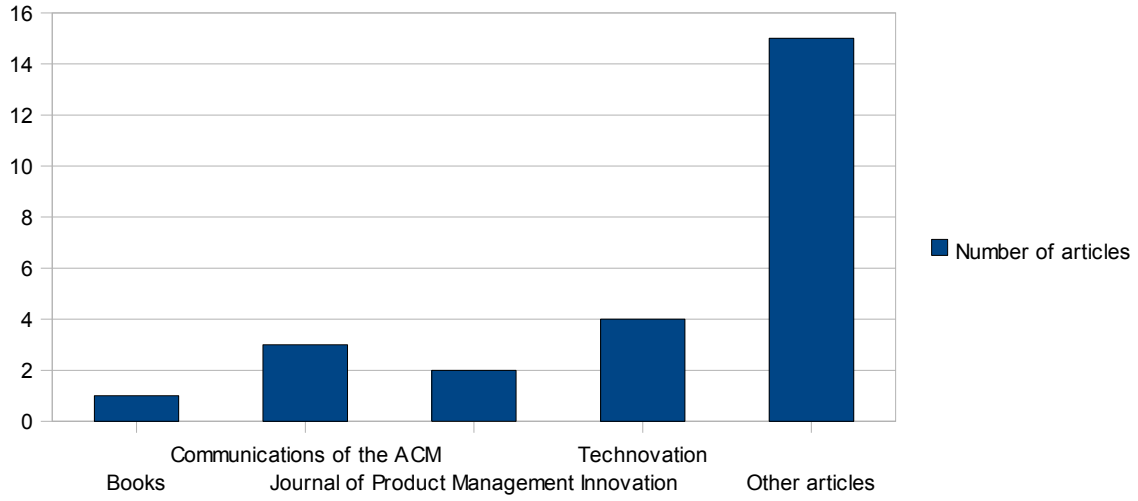


Figure 7. Articles by medium of publication