Abstract. How to apply user centeredness in the early phases of software product development? How to make the process of innovating new products more efficient and at the same time finding better fit with the product and the possible user needs? A new model is introduced for the early phases of product development of software products. It is based on a generic phase model ideas from (physical) product development complementing it with the special features of software development. The model is leaning on three inductively derived core concepts. Development processes are seen as continuing processes compared to one time design phase of physical products. The model of market development of technology adoption is used as a strategic tool. And finally, the development of a software product is seen as a process of concretising the vision of the product. The model is needed in order to start research on the interaction between different stakeholders during these early phases of software product development.

1 Introduction

In 1999, we started to collaborate with a software (later SW) product development company to understand more about their early phases of SW product design. Company has been concentrating on the user centeredness of their development process for years
and it has become evident that this user centered approach needs to be integrated fully into the whole development organization and its processes. There is a need to find a better way to support the diverse interactions between different stakeholders in the development organization. Especially, the early phases in the innovation process need to have a lot of interaction between strategic visioning, marketing and development.

**But what are these early phases of SW product development?** When do they start and when do they change to – what? The organization and its processes are not static. They seem to be changing depending on the maturity of the product and therefore the needs for interaction changes also during time.

While respecting the needs of the users more, the product development organization should pay more attention to the way it builds and shares a common understanding about the user’s activities to be supported by the new SW product. This means better management of the customer knowledge and information connected to the developing product vision and concept. But how to apply this user centered approach during the early phases of a new product? There is a need to have some kind of generic model of the early phases of product development to understand the practical interactions between the different stakeholders and the process through which their roles evolve during product development. Only then is it possible to suggest improvements for the management.

The new model introduced here is a result of an inductive conceptual analysis of the problem area. There have been ten one day common workshops between the case company representatives and the researches. A common understanding about the early phases has been build by combining the reality with some basic theoretical approaches concerning the early phases. The generalizations of the results has been but into this new model. It is concentrating on the SW products that are used by many users belonging to a certain market segment(s). This model is not for the traditional customized products even though some customer adjustments for the product might be done during some actual customer implementation.

In chapter two the book about product design and development by Ulrich and Eppinger (2000) is reviewed. It gives a good picture about product development in general. They point out the different roles people have during product development (in it’s design phase). But how will these differ with SW products? Therefore, chapter 3 looks at the nature of SW development and finally chapter 4 presents a new model for the early phases of SW product development with its main background concepts.

## 2 Early Phases of Product Development

In a generic (physical) product development process the new product is developed in project teams. In the beginning of the product development process there is just an idea
of the product that in the end will become a physical product. "Product development is the set of activities beginning with the perception of a market opportunity and ending in the production, sale, and delivery of a product." (Ulrich & Eppinger, 2000, p. 2)

Ulrich and Eppinger (2000, p. 9) divide the process into six phases: planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up (fig. 1). These are the steps through which the different inputs are transformed into the product to be launched into market. Even though the process itself is not always visible and usually differs between organizations or even projects, the development project is still seen as a stepwise transformation. Different phases describe the state of the product after that phase.

Figure 1. The product development process by Ulrich and Eppinger (2000).

Before the planning phase there is some research done and technology development is followed. The development process begins with corporate objectives, the capabilities of available technologies, and available product platforms and production systems (Ulrich & Eppinger, 2000, p. 15). During the planning phase the project’s mission statement (target market, business goals, key assumptions, and constraints) is developed. Based on this the actual development project is started. During concept development phase the needs of the target market are identified, alternative product concepts are generated and evaluated, and one or more concepts are selected for further development and testing. The concept itself is a description of the form, function, and features of a product. System-level design includes the definition of the product architecture and the decomposition of the product into subsystems and components. The layout of the product, a functional specification of each of the subsystem, and a preliminary process flow diagram for the final assembly process are developed. During the detail design a “control documentation” for the product is developed. It includes the complete specification of the product and the process plans for its fabrication and assembly. Testing and refinement involves the construction and evaluation of multiple preproduction versions of the product. During the production ramp-up, the product is made using the intended production system. The transition to ongoing production is usually gradual. At some point in this transition, the product becomes available for widespread distribution. (Ulrich & Eppinger, 2000 pp. 16-18).

These can be called as the early phases of product development since after them we have the ongoing production and sales of that new, developed product idea. When considering SW products, one could easily relate SW as a very central subsystem of the
“total” product. **SW development is tightly integrated** into the process. However, when considering product development as an interdisciplinary activity the role of SW development (especially user centered approach) gets complicated. Ulrich and Eppinger (2000) look at this interdisciplinary activity as interaction between marketing, design, and manufacturing function of the organization (Ulrich & Eppinger, pp. 3-4), which are seen (among others) as **continuous functions** having different responsibilities during each of these development phases. In order to go through these early phases of the product development we need to make the SW “ready”. That is, we need to have at least a working prototype of the SW for the ramp-up prototype of the total product to be evaluated. The early phases of this total product development actually need to cover the total process of SW development.

Particular development processes will differ from the suggested generic product development process. This generic model is most like the **process used in a market-pull situation** (Ulrich & Eppinger, 2000, p. 20). The market opportunity is already identified and the product development aims at satisfying these needs. In reality several variations of this development model are possible: technology push products, platform products, process-intensive products, and customized products. One can relate SW product to three types of these products.

“In developing **technology-push products**, the firm begins with a new proprietary technology and looks for an appropriate market in which to apply this technology. … process begins with the planning phase, in which the given technology is matched with a market opportunity. The product is unlikely to succeed unless (1) the assumed technology offers a clear competitive advantage in meeting customer needs, and (2) suitable alternative technologies are unavailable or very difficult for competitors to utilize. Project risk can possibly be minimized by simultaneously considering the merit of a broader set of concepts, which do not necessarily incorporate the new technology. … the team verifies that the product concept embodying the new technology is superior to the alternatives.”

“A **platform product** is built around a pre-existing technological subsystem (a technology platform). … Huge investments were made in developing these platforms, and therefore every attempt is made to incorporate them into several different products. … platform products are very similar to technology-push products in that the team begins the development effort with an assumption that the product concept will embody a particular technology. … a technology platform has already demonstrated its usefulness in the marketplace in meeting customer needs. The firm can in many cases assume that the technology will also be useful in related markets. Products … are much simpler to develop than if the technology were developed from scratch.”

“**Customized products** are slight variations of standard configurations and are typically developed in response to a specific order by a customer. Development … consists primarily of setting values of design variables … . When a customer requests a new product, the firm executes a structured design and development process to create the product to meet the customer’s needs. … the generic process is augmented with a detailed description of the
specific information-processing activities required within each of the phases. Such
development processes may consist of hundreds of carefully defined activities.”
(Ulrich and Eppinger, 2000, pp. 21-23)

Based on our experience with the case organization, SW product development can
be related to all of these types. SW mass product business with product families etc.
actually seems to have all these different forms of products as the target of product
development at the same time. As Carmel & Sawyer (1998) has pointed out in their
study of packaged SW\textsuperscript{1}, that the strategy is “motivated by technology-push – rather
than demand-pull (Carmel & Sawyer, 1998, p. 10)”. SW products are also considered
to be more easily customized. So the development organizations have to think also
about the platforms more carefully. Yet the demand for more usable systems points the
attention to the actual user needs – the demand-pull strategies. So there is a need to
consider the mixture of the different strategies from the point of view of process
modelling and interaction between different disciplines in SW product development.
But where to base the phase model then? Let us first look at the special features of SW
development.

3 Software Development

The well known waterfall model of the SW development describes few distinguishable
successive phases each of which has some special function to be accomplished before
moving to the next phase. This model has become the industrial standard for SW
development (ISO/IEC 12207 Software Life Cycle Processes). The successive
processes of the waterfall model are: (planning), requirements analysis, software
design, coding, integration, testing, and installation (fig. 2). After these comes the
actual use of the system – operation and maintenance.

![Figure 2. The software development process.](image)

3.1 Iterative development of software

In practice, the waterfall model has been complemented with the ideas of incremental
development and accommodation of evolutionary changes (Boehm, 1988). In order to

\textsuperscript{1} “packaged software (e.g. software sold as a product)” (Carmel & Sawyer, 1998, p. 7).
make a "big picture" also about the SW development, a classic work of Barry Boehm (1988) and his spiral model of SW development gives a good foundation. It is still valid and a good reference for the iterative development of SW based on this classic waterfall model.

The basic idea of Boehm’s spiral model was to present a framework for SW development strategy selection at some development situation. It still contains the waterfall model as a similar sequence of steps in each of its possible development cycles. Each cycle of the spiral includes: the identification of the objectives, alternatives and constraints; the evaluation of alternatives and identification and resolving of risks; developing and verifying the next-level product; and the planning of next phases (Boehm, 1988, p. 65). It is a risk-driven approach to the SW development. During each cycle of the SW development the possible sources of project risks (e.g. areas of uncertainty) will be identified. If there seem to be risks the next step should involve the formulation of a (cost-effective) strategy (e.g. prototyping) for resolving the sources of risk. Each cycle is completed by a review. (Boehm, 1988)

Boehm is concentrating on the traditional, customized system development. However, the idea of iterativeness can easily be related also to the SW product development. It is applicable both for the iterative development of the initial solution as for the continuing evolution of the SW. "The model reflects the underlying concept that each cycle involves a progression that addresses the same sequence of steps, for each portion of the product and for each of its levels of elaboration, from an overall concept of operation document down to the coding of each individual program" (Boehm, 1988, p. 65).

Boehm talks about areas of uncertainty as possible sources for (development) project risks. As long as there are some risks to be found the following work in the project is concentrated to solve the sources for that risk – resolving the uncertainties. From the point of view of the earlier mentioned demand for strategically diverse SW product development approaches and user centeredness, this gives a totally different view for defining the different phases of development process comparing to the generic model of Ulrich and Eppinger (2000).

"The spiral gets started by a hypothesis that a particular operational mission could be imposed by a SW effort. The spiral process then involves a test of this hypothesis: at any time, if the hypothesis fails the test (for example, if delays cause a software product to miss its market window, or if a superior commercial product becomes available), the spiral is terminated. Otherwise, it terminates with the installation of new or modified software, and the hypothesis is tested by observing the effect on the operational mission. Usually, experience with the operational mission leads to further hypotheses about software improvements, and a new maintenance spiral is initiated to test the hypothesis."

In SW product development this spiral model should be elaborated towards more evolutionary development. This means that instead of the uncertainties of one “operational” system we should consider the uncertainties of our product idea. The
work of the product organization should somehow concentrate to solve the sources of uncertainties (or risks) of product’s market penetration through evolutionary development steps.

3.2 Staged development of software products

After the initial development of the SW at least some form of maintenance development is needed or even during the initial development different prototyping techniques have been used. Rajlich and Bennet (2000) have named this as ”staged model for the SW life cycle”. Here SW development is seen as a series of distinct stages with different activities, tools, and business consequences. Their model of SW lifecycle consists of five distinct stages: initial development, evolution, servicing, phaseout, and closedown. During initial development engineers develop the system’s first functioning version. During evolution, the engineers extend the capabilities and functionality of the system to meet user needs. During servicing engineers make minor defect repairs and simple functional changes. At the phaseout, the company decides not to undertake any more servicing, seeking to generate revenue from the system as long as possible. During closedown the company withdraws the system from the market and directs users to a replacement system, if one exists. (Rajlich & Bennet, 2000)

Rajlich and Bennet (2000) also represent the versioned staged model, the backbone of which is the evolution stage (fig. 3). At certain intervals, a company completes a version of its SW and releases it to customers. Evolution continues, with the company eventually releasing another version and only servicing the previous version. The version refers to strategic changes during evolution, and the release refers to servicing patches.

Now, the most interesting phases are initial development and evolution. During initial development, engineers build the SW to satisfy initial requirements. From the point of view of future iterations, this stage lays two important foundations: SW team expertise and system architecture. During initial development, the team adds significantly to its knowledge of the domain and the problem. This expertise is critical for future evolution. System architecture refers to the fact that the system components, their interactions, and their properties, such as functionality and efficiency, may aid or hinder change during evolution. If the initial development is successful, the SW enters the evolution stage, when iterative changes, modifications, and deletions to functionality occur. Evolution partly results from the learning process. Sometimes companies release SW immediately after initial development, but most often the SW is released during evolution after the SW has gone through several internal iterations. The release date is also based on sometimes conflicting technical and business considerations, such as time to market, time to delivery, SW stability, and fault rates.
The model seems to be useful, but there are still a few questions. What is the basis for the initial requirements? When will we know we have succeeded with that stage and are ready to move to the next evolution stage? Is it possible to give more accurate definition for the release dates? What is the role of different marketing strategies in this model? The new market centered phase model for the early phases of SW product development tries to give answers also to these questions.

4 Market Centered Phase Model

This model for the early phases of SW product development is based on an inductive integration of the concepts from research referred earlier (phase/stage thinking, interdisciplinary, iterativeness and risk-driven strategic choice) and from the ten common workshops between the company representatives and researchers (continuing processes, technology adoption and concretising vision). After explaining the three main concepts derived from the workshops the first version of the model is presented.
4.1 Continuing development processes

During the workshops it appeared that the traditional SW lifecycle processes (e.g. planning, requirements analysis, design, etc) are more like continuing and parallel processes of the product development organization (like marketing, design and manufacturing in Ulrich and Eppinger (2000)) rather than chronological phases of production. They can be seen as levels of hierarchy for each new SW version. Every time the organization makes new version or release of the product, it is doing all these life cycle activities (or at least choosing among them) but with different emphasis. While moving to a design phase of some product version, they are still continuing with the requirements for another version. SW development is iterative among these continuing processes.

For the first developing version of this model these processes has been grouped into four levels of production (fig. 4).

**Figure 4. Software development processes and the iterative nature of software development.**

- **Visioning** is the process through which we concretise our idea of the product in order to find a solution for some broken business process(es) and to find the potential market and the possible users inside that market. We declare some generic services for the product and build a shared understanding about the context of use, key assumptions and our business goals.

- **Concepting** is the process through which the different needs and requirements are evaluated in order to concretise the product features for the SW product.

- **Development** is the traditional design, coding and testing of the selected product features that are going to be included into the product. During **Implementation** the final customer solution for some specific customer is constructed. When compared to the generic product development model these are like the continuing processes of marketing (visioning), design (concepting), and manufacturing (development and implementation). But where to get the phases for development then? For this we introduce the technology adoption model.
4.2 Technology adoption model

High-tech product development is based on the idea of discontinuous innovations. This means at least two things. First, the development of new products is very often more technology than demand driven. Second, while introducing new products we must also consider the change that these products are causing for the current way of working for their users. Because product development business is reaching for mass markets, attention has been given to the possible ways users are accepting these new high-tech innovations. Geoffrey Moore has used the model of Technology Adoption Life Cycle (with his “Chasm” theory) as an approach for high-tech marketing. It describes the market penetration of new products “in terms of a progression in the types of consumers it attracts throughout its useful life” (Moore, 1995, p. 11). The consumers are grouped into 5 categories: innovators, early adopters, early majority, late majority and laggards. Technology adoption life cycle looks like a standard deviation curve (fig. 5), where early majority and late majority fall within one standard deviation of the mean, the early adopters and the laggards within two, and at the very onset of a new technology are the innovators (Moore, 1995, p. 12). This model has also been taken here as a starting point for understanding the different developmental phases of user or market centered SW product development - especially the early phases of product markets. It is like a strategic tool for business visioning and management of product development organization, not a purely explanatory model of evolution.

Figure 5. Technology adoption lifecycle model. (Moore, 1995)

This market perspective can be used for understanding users and their activities when organizing product development into different stages during time. The vision of the new SW product – and/or the need – is changing with the development of the user’s activities and developing view of the product itself. SW development is actually concretising the vision.

4.3 Software development as concretizing its vision

The third concept is based on the idea that SW product development is iteratively developing the vision of the product into more concrete form (fig. 6). First we have an idea of the product to be used in some abstracted activity of some (segment of) users. That is our tool for product development. We make a version or release of the product
Product development looks for the desired SW features to be used in the user activity. It abstracts the individual wishes in order to serve others with some similar but also slightly different purposes. It prioritises the different possible features based on the current view of market potential. This is done through different interdisciplinary activities in the development organization. People are processing some work products using different conceptual and SW tools for the work. One important conceptual tool is a shared understanding of the abstracted user activity for which the product is being developed.

The vision of the abstracted user activity is both enabling and limiting tool: it empowers the people in the transformation process with the historically collected experience and skill “crystallized” to it but it also restricts the interaction to be from the perspective of that particular tool only – other potential features remain “invisible” (Kuutti, 1994). That is why there is a need to make pilots to be tested in some real context. The pilot is also a possible trigger for further changes. By testing the pilot in real and developing use context “other potential features” – e.g. a new vision of the developing system – may become visible.

4.4 Early phases of software product development

The new phase model in this article leans on the market penetration viewpoint. The purpose of the early phases of SW product development, based on the Moore’s model of technology adoption (Moore, 1995, p. 17), is to define the right customer segment for getting the interest of the majority. The actual stages of these early phases are defined by a risk-driven strategic choice of what is needed from the product to be able
to test and concretise its vision. It is also a point in which the decisions are made about moving to the next phase. The actual release (or version) of the product is a result of the continuing processes described in 4.1. The implementation of this product for some user is a tool for its evaluation. This evaluation is clearly an interdisciplinary activity.

At the management level the early phases form a process through which the idea of the product gets elaborated and new information is gained for strategic decision-making. The interaction between the continuing processes (4.1) might be structurally the same during each phase, but the roles and responsibilities are depending on the phase and changing and evolving with the main phases. Also the inputs and outputs (or at least the scope of content of some formally defined work product) might vary from phase to phase. Development of a new product usually starts as a project of just a few people. During the later phases with much more mature product the development relates more to the standard processes. In between there is a mixture of roles and responsibilities. During the first phases the project manager has more responsibilities and later the role of product manager appears. We continue with the development cycles (4.1) until there is enough information to make strategic business decisions to move to the next phase. Part of the moving to the next phase is also planning of the roles and responsibilities and their transition. So the organization of the product development is dynamically changing and evolving with the developing product. Next is an illustration of what kind of phases there could be.

4.4.1 Vision pilots

First we have some idea of the possible future product or idea of the possible needs or services in the future for which we think we might find the best solution. We need to become sure that the idea is technologically feasible, that it is possible for human beings to use etc. After that we can define our strategy for the idea and start to identify the potential market segments. During this phase we need to make certain parts of the idea more concrete – developmental pilots – to be able to evaluate and test some parts of the idea. We also work out our presumptions about the possible use of the system and its markets. We might need to do these “tests” more than once. At the management level we use this phase to make the decision to start developing the product for real users of some possible markets, e.g. start developing segment pilots.

4.4.2 Segment pilots

Now we know some limits of our idea, and can start to look for possible future markets and their special needs. For this we implement certain features of our idea for certain user activities to try on. By comparing several implementations we start to see the market potential of our idea. We carefully consider what is needed for these kinds of implementations. What kind of results of these tests we need in order to define the segment for our first target. What are the needed user or use activity tests. How long
should they take, what amount of functionality we need to implement and when. What are the phases through which the possible target segment needs to go in order to be able to make the transition from old to new? We might need to do this for several possible user activities (customers or customer sets) in order to be able to concretise the special characteristics of the new system and the separating features of the possible first target segment. We also start to separate the core product features from the customer centric features. At the management level we use this phase to make decision to start developing the product for our first target market segment, e.g. start developing first target release.

4.4.3 First target

When we get the first target release of the product ready for the best possible market segment, we start to look for the next possible market segment or the appearing new possibilities for strengthening our current market segment. We focus certain customer implementations as test cases for further development. We need to reconsider again what is needed from these kinds of tests. As earlier, we need to do this for several users (now outside our first target segment) in order to be able to concretise the special characteristics of our new release. We also need to collect the feedback from the first target users. From the product point of view we try to define as many common features as possible to strengthen our core product and to define more precisely the special segmented features. Our goal is to do finally as little customisation as possible. At the management level we make a decision to go to the next phase, e.g. start developing next release.

5 Conclusions

There is a tendency to think that spending too much time with one or few customers is not efficient enough for SW product development where the focus should be more on the abstracted common features for the whole customer segment and representative cases. This may be true if we think about the customer as only a source for possible new SW features. But at the same time, if we give ready-made products out for some test bed markets, and find no financial opportunities with it, that cannot be very cost-effective either. We need to take the user/market viewpoint seriously while prioritising different features and their timing.

This market-centered model is an attempt to model the early phases of SW product development. The model connects time, SW lifecycle, market penetration, and customer implementations into one “interactive” whole. The phases of product development are there to get a better picture about the user activities (of majority) and a better solution to support these activities. During the development people share an
abstracted view of that user activity and are actually both concretising that view and building the product to fit this developing view. To manage this naturally expanding process of product development we need to manage the interaction inside the (also evolving) product development organization.

Product development activities of SW product are actually merging the design and manufacturing processes of a physical product together. The early phases of development should therefore be based on other criteria than “design”. The models for SW development, even iterative, are built on the idea that the requirements “are there” to pick up and fix the problem. But for the new product innovation the situation is not that clear and the business might even need to “grow” these requirements. So there is a need to concentrate more on the lifecycle of the innovation, not the development as the earlier models suggest. This new phase model still valuates the stepwise approach from the earlier models but gives (at least for the case organization) a better framework to reflect the actual work of SW product development. However, the model in this article is not yet focusing on the interaction between different stakeholders and this is where it is developed in the future.

6 Acknowledgements

This work has been done in the Finnish national KESSU project financed mainly by the Finnish Technology Development Center, TEKES. I would also like to thank Pirkko Jokela and Eija Suikola from Teamware Group Oy for challenging theories from their practical viewpoint. Special thanks goes to professors Kari Kuutti and Samuli Saukkonen who have significantly contributed to the development of this model.

References