TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

Bridging the Gap between Management and Engineering: An Extension of the V-Modell XT Framework

Ausgeführt am Institut für

Softwaretechnik und interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von Prof. Dr. Stefan Biffl und Dipl.-Ing. Dietmar Winkler

durch

Franz Stimpfl

Pfarrer Karl Hesslegasse 32
3002 Purkersdorf

25. September 2007

Abstract

In today's business world, a world where globalization, the optimization of processes and workflows, and the maximization of profit is commonplace, the basic question 'Did the customer get what he wanted?' is still not sufficiently solved.
Several well known quality standards, such as ISO 9001 or the Capability Maturity Model, have been developed and it has been observed, that every company has developed their own strategy for planning and developing new products. Despite all of these quality standards, products are still postponed or even canceled. With the result that customers often have to cut back on their expectations and requirements and often remain unsure to get the desired product.

This diploma thesis focuses on the difference between management and engineering. These two disciplines use different models, which obviously have different fields of expertise and due to this specialization use a different language each with their own terminology and idioms - especially engineering.
Management, for example, uses models like the Boston Consulting Group Matrix or a Strengths Weaknesses Opportunities and Threats (SWOT) Analysis. These models are located in the business layer of the company. Using these strategic models, management tries to lead the company into a direction which achieves a maximum of profit for the company.
Engineering, on the other hand, uses models such as the V-Modell XT, the Software Process Improvement and Capability Determination model (SPICE) or SCRUM, which are located at a lower, technical layer of the company. These models are employed especially during development and reside mainly at the project-level.
Mostly the differences in the vocabulary used and their different focus easily lead to misunderstanding between management and engineering. The problems in successfully communicating often manifest themselves in delayed products or products that fail to fall within the desired specifications.

The goal of this diploma thesis is to combine management and engineering processes in using one model. Therefore a meta-model, which is then mapped onto an existing process model, is developed. This meta-model has defined processes and defined interfaces between management and engineering. There are also defined requirements and deliverables for both - management and engineering.
One of the current models is the V-Modell XT, which on one hand already has protocols

on how a software-system should be developed on the project-level but on the other hand in the current version has no mechanisms for product management, is chosen to map the developed meta-model to. Both models are broken down into small phases, and the roles, products and activities of each phase are analyzed and compared to determine their equality or to point out their differences. The result is an extension to V-Modell XT, which completely covers the product planning process.

The second part of this diploma thesis covers the application and integration of the proposed extension into already existing processes and projects of a small development team. The application and integration are tested in a case study, which is performed in a software development project in an academic context. As a result, the team and also companies, who use this extension, would have one model, which covers the whole procedure, starting from the idea for a new product to the final rollout. All these steps are covered in this model and ensure a certain level of quality.
The main benefit for companies using this extension of the V-Modell XT is, that they can continue to use their well established development models, for example Extreme Programming, apply a mapping model to their projects, and therefore are able to make their projects V-Modell XT compliant.
An additional benefit of this model is the simplification of communication and as a result a decrease of misunderstandings within a company.

While this diploma thesis focuses on the software industry which chronically suffers from delayed products and to which the resulting model is applied to, the proposed extension can easily be transfered to other industries. Every company, which has a product management or a product management department, are most likely to have their own product management model and therefore will be able to use this extension to the V-Modell XT.
The quality managers, software developers, project managers or product managers are the target audience of this diploma thesis. They are the ones who adapt and establish this extension in their companies, to further assure the development of excellent products, which are delivered on time and within budget.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzen Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Franz Stimpfl, 9925085

Contents

1 Introduction **2**

2 Management Models **10**

2.1 Product Management 10

2.1.1 Product Planning 12

2.1.2 Product Analysis and Strategies 15

2.2 The Product Life Cycle 22

2.3 Product Development Process 25

2.3.1 ABB Gate Model 25

2.3.2 Siemens prodSEM 28

3 Engineering Processes **29**

3.1 V-Modell XT 29

3.2 Unified Process 35

3.3 Agile Development Methods 47

3.3.1 Scrum 47

3.3.2 Agile Unified Process 51

3.3.3 Extreme Programming 53

3.4 Comparison of the presented Engineering Processes 57

4 Research Approach **59**

4.1 Research Goal 59

4.2 Solution approach 62

4.3 Evaluation 63

5 Product Development Process Meta-Model **67**

5.1 Overview 67

5.2 Phases 69

5.3 Decision Gates 70

5.4 Meta-Model Summary . . . 71

6 Integration of the Meta-Model into the V-Modell XT 73

6.1 Meta-Model compared to V-Modell XT . . . 73
6.2 The PDP Extension . . . 74
6.3 Additional Roles and Products . . . 76
6.3.1 Roles . . . 76
6.3.2 Products . . . 76
6.3.3 Activities . . . 77

7 Evaluation of the Extension 79

7.1 Project Description . . . 79
7.2 Operation . . . 83
7.2.1 Preparation . . . 83
7.2.2 Execution . . . 83
7.3 Results of the Case Study . . . 84
7.3.1 Hypothesis 1 - Feasibility . . . 85
7.3.2 Hypothesis 2 - Small Project Teams . . . 86
7.3.3 Hypothesis 3 - Agile Development Process . . . 87
7.3.4 Hypothesis 4 - Successful Project . . . 89

8 Discussion 90

8.1 Hypotheses Analysis and Interpretation . . . 91
8.2 Research Question Analysis and Interpretation . . . 93

9 Conclusion and Outlook 94

Bibliography 99

1 Introduction

In the last years the consequence of globalization and convergence of markets are more and more effecting industries. Thus the optimization of processes and workflows and the maximization of profit has become a ever more challenging task for companies regarding the global and emerging markets. Currently the industries and even more the highly volatile software industry has to anticipate the rapidly changing technologies. A manifestation of a crisis in the software industry during the 1960ies spawned the area of software development processes. But the basic question, which can be expressed as ‘Did the customer get what he wanted?’, still remains to the present day.
Several well known quality standards, such as ISO 9001 or the Capability Maturity Model, have been developed to address this problem and it has been observed, that every company has developed their own strategy for planning and developing new products. Despite all of these quality standards, software products are still postponed or even canceled. With the result that customers often have to cut back on their expectations and requirements and often remain uncertain to get the desired software product.
This issue is not completely new to the software industry, already in 1979 David L. Parnas described the problems of software development. Parnas addresses some statements of software industry in [57].

’We were behind schedule and wanted to deliver an early release with only a proper subset of intended capabilities, but found that subset would not work until everything worked.’

’We wanted to add simple capability, but to do so would have meant rewriting all or most of the current code.’

’We wanted to simplify and speed up the system by removing the unneeded capability, but to take advantage of this simplification we would have had to rewrite major sections of the code.’

A trend analysis [76] performed by the consulting company Infora in cooperation with meetBIZ & Denkfabrik states, that in the last two years only every fifth software project is completed according to the requirements. As a reason, Infora states, that the changing requirements during the project and lack of project management cause the troubles during the project. This study shows, that the topic is still present. The next passage will give several examples, which are the result of different approaches to software and product development.

An example of working respectively non working management models can be taken from the gaming console industry. The competitors Sony and Nintendo, with their products Playstation 3 and Wii. Sony chose to use the cell processor and even developed the Blue-Ray technology for its PS3. At the end of 2005 Sony stated, that the console would be available only for the japanese market in spring 2006. For the US market Sony stated, that the PS3 would be available fall 2006. At first the price was estimated to be 300 to 400 USD. Today the actual costs for the PS3 are 599 USD/EUR. The Wii on the other hand did not focus on the newest technology. Also the costs for the Wii, with 249 USD/EUR, are much lower than the costs for the Playstation 3 [72].
Were the requirements for the PS3 too special? Did Sony overestimate the requirements for the PS3?

Another rivalry for many years now is the ongoing competition between AMD and Intel. It is a fact, that AMD has the better processor design, than their Intel counterparts. If AMD has the better processor performance [85], why does not everyone buy AMD processors? Maybe many customers do not know about this fact? Is this information made available to the customer at all? The average user does not have the necessary technical background. Is Intel's marketing better than AMD's? Is there a lack in the product management process of AMD? Did they cut costs at the wrong position?

Microsoft's management has to deliver a product - Windows Vista - even if it is not completely finished and still has major bugs, just to have a new product on the market on time. This example shows, that it is quite common for management to dictate the requirements for a new product, whether they are feasible or not. Is there a mechanism to control management in its actions? Are there procedures, which have to be abided to follow a company wide process? If there is an established process, is top-management responsible for its faults?

Apple's iPod is another good example for successful product management. While Apple did not invent the portable MP3 player, the iPod is often used synonymously with portable music players. The questions that are raised here are:
Why is Apple this successful with their product iPod? Is there a special mechanism or process for this?

These examples show that companies exist, which have proven to be successful in finishing software projects in time and in managing the development of new products. But why is it still not possible, to solve this problem in general and finish every software project on time and within budget?
This question is often difficult to answer. Most of the time all parties involved in a software development project are certain, that they understood the customers correctly and that everybody else is wrong about the outcome of the product. Besides the trouble to get the user to realize what he wants or what he really needs, there is also a major difficulty within a company - the difference between management and engineering.
This very problem is nicely illustrated in Figure 1.1, where the different views of a product are depicted. It can be seen, that everybody involved in the project, from management to developer or even the customer, must have very different ideas about the vision of a product. Management and engineering have different models and processes, and as a result use different vocabulary. Not only the difference in vocabulary, but also the lack of communication respectively the lack of qualitative communication is essential for the failure of software projects.

Overbooking of development resources is common and without a big picture and everybody knowing what to focus on, the efforts cannot be systematically replanned or reprioritized. Instead the developers have to decide which activities to focus on and which activities to put on hold. Hence important tasks are not finished as planned. The implications of the decisions made in a project, cannot be estimated because of the missing overall view and the unknown resource interdependencies. Prioritization 'happened' through fire-fighting or personal judgment and are a result of missing communication within a company. Decisions are made on project level, only considering the implications for this current project, not for the entire portfolio.

Some software development companies up to now do not manage their development ac-

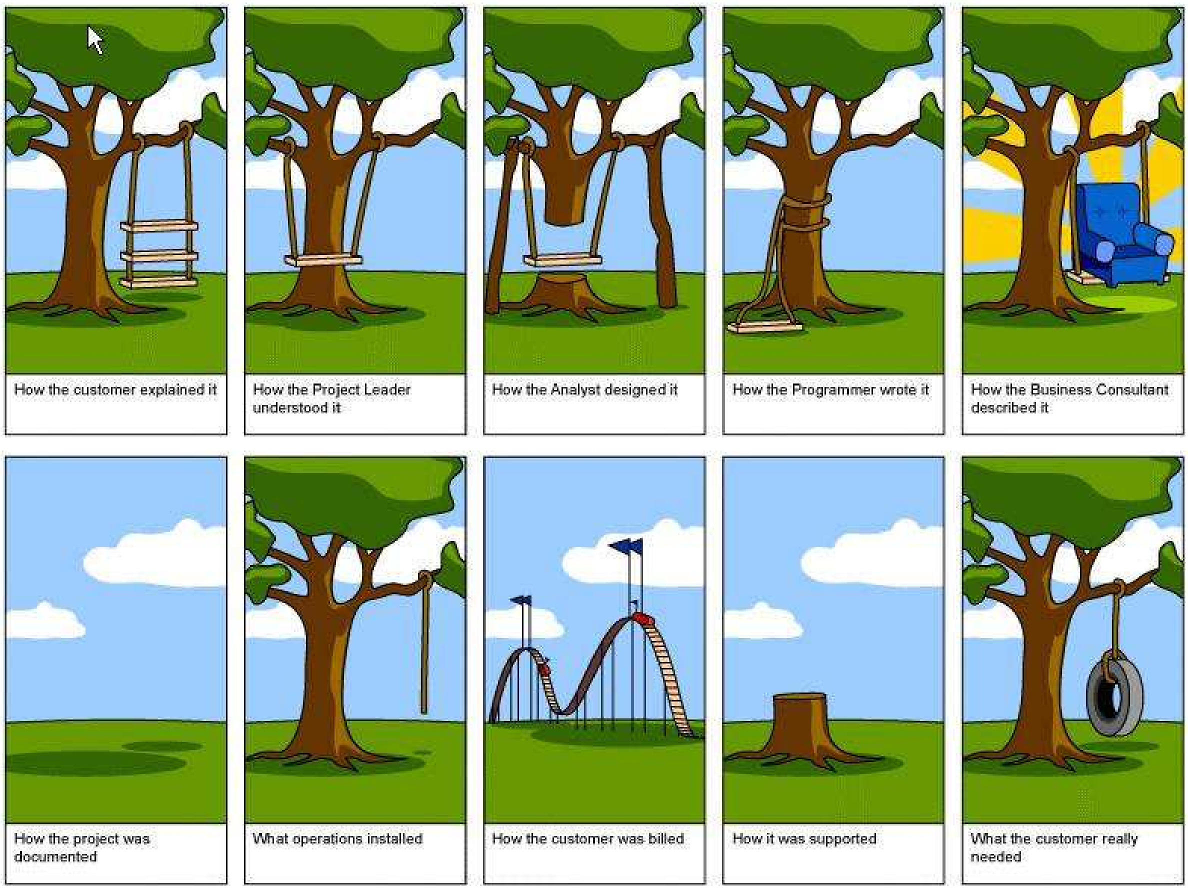

Figure 1.1: What the customer wanted. [71]

tivities using portfolio management. They lack clear priorization of their development effort. Starting new projects is considered both frequent and impulsive in nature, often without finishing old assignments or deliberately put on hold. Because different activities are not necessarily explicitly managed as projects, or even recognized as part of the product development portfolio, resource planning is seen as difficult, and often is omitted entirely because of its perceived futility. There are no explicit lists of features, which have to be implemented, but rather ideas for future improvement. These improvement resided in various text files or mails. The area, which often needs the greatest improvement, is to provide a common understanding of what projects and other development activities are underway, how these are resourced, what their relative priorities are, and why they are conducted.

This diploma thesis focuses on this very difference between management and engineering. These two disciplines use different models, which obviously have different fields of expertise and due to this specialization use a different language each with their own terminology and idioms - especially engineering.

This situation is depicted in Figure 1.2. It can be seen, that there is a big barrier between management and engineering, which, in some cases, keeps them from successfully finishing projects.
Management, for example, uses models such as the Boston Consulting Group Matrix or a Strengths Weaknesses Opportunities and Threats (SWOT) Analysis. These models are located in the business layer of the company. Using these strategic models, management tries to lead the company into a direction which achieves a maximum of profit for the company.
Engineering, on the other hand, uses models such as the V-Modell XT, the Software Process Improvement and Capability Determination model (SPICE) or SCRUM, which are located at a lower, technical layer of the company. These models are employed especially during development and reside mainly at the project-level.
Their differences, in used vocabulary and their different focuses, easily lead to misunderstanding between management and engineering. The problems in successfully communicating often manifest themselves in delayed products or products that fail to fall within the desired specifications.

The motivation of the first part of the diploma thesis is the question, if it is possible, that a company can use only one model for product development throughout all departments. 'Is it possible, that management and the engineering use the same model?' The benefits would be, that only one model has to be established in the company. Every employee has the same vocabulary within the model.
The goal of this diploma thesis is to combine product management and engineering processes by using one single model. Using this model managers can keep track of the development process and software and system developers know what the main features of the new product should be and can concentrate on them.

Therefore a meta-model, which is then mapped onto an existing quality model, is developed. This meta-model has defined processes and interfaces between management and engineering. There are also defined requirements and deliverables for both - management and engineering. One of the models, currently in use in the field of quality assurance, is the V-Modell XT. This model on the one hand already has protocols on how a software-

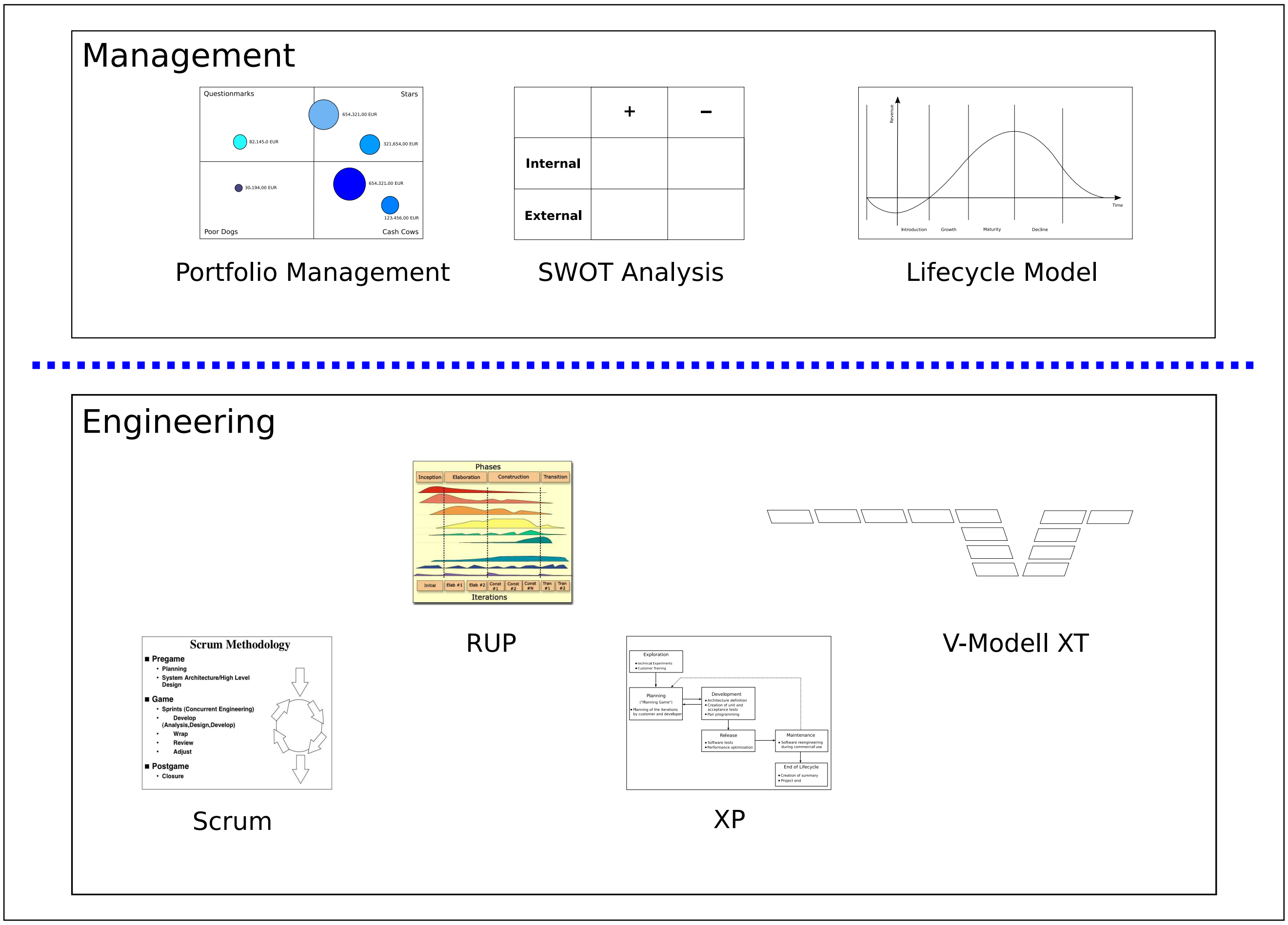

Figure 1.2: Research Goal: Up to now management and engineering use their models only in their field of expertise with only a little input from other departments. This creating a gap between both disciplines.

system should be developed on the project-level but on the other hand in its current version includes no mechanisms for product management. It is this model, which has been chosen to map the developed meta-model to. Both models are broken down into small phases and the roles, products and activities of each phase are analyzed and compared to determine their equality or to point out their differences. The result is an extension to V-Modell XT, which completely covers the product planning process.

The second part of this diploma thesis covers the integration and adaptation of the proposed extension into already existing processes and projects of small companies. As a result, these companies would have one model, which covers the whole procedure, starting from the idea for a new product to the final roll-out. All these steps are covered in this

model and ensure a certain level of quality.
The approach proposed in this diploma thesis is a new extension to the V-Modell XT which combines these two disciplines with clearly defined interfaces between management and developers. Using this model, managers can keep track of the development process and software and system developers know what the main features of the new product should be and can concentrate on them.
For small and medium-sized enterprises (SME) the use of development models should be easy and cost-effective. It should be possible for a group of only a few people to be able to manage a project according to the selected model. The problem for SME is, that they often do not have enough employees to cover all tasks, which need to be preformed when using a certain development model.
Due to the wide range of this topic, delimitations have to be set. Though Product Management is tightly coupled to Project Management, this diploma thesis does not cover Project Management at all, neither does it deal with Marketing and Sales.

This diploma thesis can be divided into the following parts:

1. The presentation and analysis of current management models:
   This part presents several existing models, in the business level, which the extension of the V-Modell XT is based on.

2. The presentation and analysis of current development processes:
   During this chapter development processes are presented and their advantages and disadvantages are pointed out.

3. The research approach:
   The research questions and the main focus of this diploma thesis are analyzed and presented in this chapter. Also the hypotheses for the evaluation of the created extension are presented.

4. The creation of a product development planning meta-model:
   This chapter covers the creation of the meta-model. Information gathered in the last part is extracted and formed into a meta-model. This meta-model covers the product planning process. It is used to test the V-Modell XT extension against.

5. The integration of the meta-model into the V-Modell XT:
   In this chapter the V-Modell XT is mapped to the meta-model. Similarities and differences are extracted. These differences are further described and form the new extension.

6. The evaluation of the new extension of the V-Modell XT:
   In this chapter the application of the newly created extension is tested. This test is performed in a case study, where a product is created by a small group of developers.

7. The discussion of the results from the case study:
   The results of the case study, which have been presented in the last chapter, are discussed in this chapter with regard to to the formluated hypotheses.

8. And finally a conclusion and outlook:
   This chapter summarizes the whole diploma thesis and gives an outlook over possible development for the future.

2 Management Models

The following chapter will present the management aspect of the business area this diploma thesis is located in and review some of the many models and processes used in today's business world.
As the introduction already outlined, one topic of this diploma thesis covers the Product Management discipline. In order to clarify the term Product Management, the next chapter gives a short introduction.

2.1 Product Management

What is product management? [5] defines product management as '... *the discipline of planning, development, production, marketing and disposal of a product for the best possible outcome for supplier and customer'*. A lot of other different definitions can be found. It is obviously not easy to generally define product management.

The discipline of Product Management is well known to consumer industry. The birth of modern Product Management is the 13. May 1931. Shortly before that day, the head of the marketing department Neil McElroy from the company Procter & Gamble (P&G) was asked to take care of the launch of a new soap-product 'Camay'. But the success of the already established branch 'Ivory' should not be at risk. McElroy proposed a memorandum with the mentioned date, that he should not only be responsible for marketing of the new soap but he should rather be the head of a new one-product-only company outside of the usual marketing-department responsible for soaps and also should he be responsible for all other product related activities and therefore be responsible for the success of the product 'Camay'. The former president of P&G, Richard R. Deupree, soon was convinced, so that this concept was also taken for all other products within the company. The main idea behind this concept is, that using custom support for all, also concurring, products, the market success of the company can be increased. Since then the idea was wide spread, but there still are companies, which think that they can go without actual product management. [78, 79]

The role of a product manager varies widely, depending on the company, but there are several key responsibilities, a product manager is in charge of. These responsibilities can be grouped into six main categories:

- Market Research
- Product Definition and Design
- Project Management
- Evangelizing the Product
- Product Marketing
- Product Life Cycle Management

Market research is the task of examining and gathering all aspects of business environment. Factors, such as trends in the market, market structure, competitors, government regulation and so on, are investigated. Market research is very helpful while creating a business plan, planning a new product or expand to new markets. Basically a market research helps companies to make better business decisions about the development and marketing of new products.

A product manager is responsible for the definition and design of the new product. Based on a market analysis, a results of research and development or customer requests, the new product will defined and afterwards created.

For a successful completion of a project, the discipline of project management is substantial. Project management deals with coordinating, allocating and scheduling resources in a way, that projects are finished in time and within budget.

Another important task is evangelizing the product. It is living the product.

Product marketing important for distributing the advantages and features of the new product. Product marketing includes the tasks which are commonly known as marketing. Presenting the product to the consumer respectively companies at fairs or on radio, television and print media, is included in product marketing.

The product manager needs to keep the normal product lifecycle in mind, to plan further updates to the current product, or, if the product is not profitable anymore, end it's lifecycle.

Product management is often used synonymously to all the tasks, which have to be performed, to deliver a particular product to the market. This huge amount of tasks can further be divided into two main categories - **product planning** and **product marketing**. Product planning subsumizes all planning activities while product marketing combines all tasks of outbound messaging - telling the world about the product. As already mentioned, depending on the company, a product manager has a different focus in these categories, which may be one reason, why product management is quite hard to capture.

The following section will shortly outline successful product planning. The extensive task of product marketing will not be discussed, because this would go beyond the scope of this diploma thesis.

2.1.1 Product Planning

As already mentioned, product planning is one of the main tasks of product management. It covers the search, the review and the selection of promising ideas for a new product. Also the definition of this new product as well as several preparation tasks, such as

- investigation of product-influences,
- identification of product-causes,
- performing of product-analysis and
- development of special product-strategies

have to be done in this phase. The implementation of the given measures as well as the following gathering of ideas and definition of the product are tasks, which are normally performed by a small group of people in a rather short period of time. On the contrary development of a new product is in most of the cases at least ten times as much effort. But the most important fact is, that product-planning, in relation to the effort spent, has the most influence on the outcome of the product, compared to all other tasks. Planning is at the beginning of the product-creation process and it is greatly responsible for the product-development costs of the following phases and of course the success of the product. (cp. [5])

2.1.1.1 Product Influences

Product planning has the most influence on the product, but what are these influences? According to [5], the influences can be divided into company-internal and company-external influences and are listed in the following:

Company internal influences are:

- the degree to which the company is customer oriented - Are management, employees and processes focusing on customer needs?
- the organizational structure of the company - Does the company have a flat or deep hierarchy?
- the integration of product management in the company - Is product management an established discipline in the company?
- the experience of employees - To which degree are the employees knowledgeable about the field they are working in?
- how the company deals with know-how - Is there e.g. a knowledge management system? Are there established procedures for knowledge management in the company?
- the financial situation of the company - Does the company have resources for further expansion or research?
- the company's culture regarding leadership and readiness for cooperation - Are people ready to help each other or just focused on their own career?

On the other hand the external influences:

- the different kinds of markets, their evolution, current status and development possibilities
- the customers - Are customers willing to buy the new product? Do they need the new product?
- the competition - Is the market highly competitive?
- the suppliers - Are the supply chains working? Will there be an increase in price?

- the technology - Is the technology available to me? Are there others who have the same knowledge? Is it a market entry barrier?
- the power and potential of the economy - What will the reaction of the economy be?
- politics - Will there be political controversies started with the introduction of the new product?
- local, national and international laws - Is the product according to current laws?
- the workers' council - Do my employees have enough rights?

To indicate internal and external influences and their degree of their influence, to rate and analyze them is one of the most important tasks in product planning. (cp. [5])

2.1.1.2 Product Triggers

Possible triggers for planning a new product could be one or more problems, one or more opportunities or a combination of both. Typical problems in this context are:

- complaints or new requirements from the customer - Is the current product working? Did the company meet the requirements of the customer?
- changes in the market - Is there a new competitor or new technology?
- sales or profit loss - Is the company able to fund its further existence?
- changes at the suppliers - Are the suppliers in the current form still present, e.g. bankruptcy or new delivery-conditions?
- political or economic crisis

On the other hand opportunities arise mostly from:

- new technologies - 'The chance to put the money on the right horse' or establish a new market for the new technology
- internal or external patents - Are the researches of the company limited by certain patents?
- new markets or new conditions on the market - e.g. Are there new laws? Did the competitors catch up?

- synergies because of cooperations or company-takeovers - Maybe a strategic alliance will strengthen the position against the market leader?
- pioneer-products - Pioneer products are the best possibility for a company to become the market leader. It is highly risky, but if it succeeds, it is highly lucrative.

While searching for new product ideas, it is not enough to know the influences and triggers for a new product, but it is essential to know the link to the own product program. It is important to know the positioning on the market, the strengths and the weaknesses in comparison to the competitors. (cp. [5])

2.1.2 Product Analysis and Strategies

The following section presents a few widely used strategies to analyze new products and product portfolios. The main goal of these strategies is to optimize the way a company invests it's development resources, to achieve an optimal revenue.

Portfolio Management

A product portfolio is an overview of products which are currently in development or which will be developed in the future. These projects share certain similar goals, resources or ideas. Portfolio-management is the task to find the similarities of projects and efficiently assign development resources.
According to a best practice study [81, 82], Cooper states that portfolio management is a critical senior management challenge and top performing businesses tend to rate the importance of portfolio management much higher than poorer performing companies. He explains this state using three facts [80]:

1. The success of a new product is the foundation of the success of the business. Therefore, the ability of selecting a project today's projects to become tomorrows product winners.
2. The development of new products is the implementation of a company's business strategy. If the new chosen product ideas are wrong, the projects are wrong respectively the project balance is wrong and therefore the company fails in implementing their business strategy.
3. Portfolio is about resource allocation. In today's business world, where everyone has to do more with less resources, the consequences of poor or missing portfo-

lio management is the scattering of scarce resources and as a result, starve those project, who would deserve being implemented.

Cooper and Edgett [80] further define four goals in product management. These four goals are presented in the following.

1. Maximize the Value of Your Product: This goal aims to select new development projects to maximize the sum of values or commercial worths of all active projects in the pipeline in terms of some business objective.

2. Seek Balance in Your Portfolio: Here the goal is to achieve a desired balance of projects in terms of a number of parameters, such as long term projects versus short ones, high risk versus lower risk projects. This technique mostly uses graphical representation of the data, to better portray the balance.

3. Your Portfolio Must Be Strategically Aligned: This goal means that all the projects are 'on strategy' and that the breakdown of spanning across projects , areas, markets, etc., must mirror your strategic priorities. There are several portfolio methods, designed to achieve a strategic alignment, such as the Boston Consulting Group matrix, which is described in the next section.

4. Pick the Right Number of Projects: Often companies have too many projects underway for the limited resources available. The result is pipeline grid-lock: projects end up in a queue, they take to long to reach the market, and key activities are omitted because of a lack of people and time.

Portfolio management is the most popular method and is used in nearly all companies as the favored tool regarding investments. As an addition to portfolio management the strength-weaknesses analysis can be used. This technique was developed by Albert Humphrey at the Stanford University in the 1960s and 1970s.
Patzak and Rattay points out several benefits of portfolio-management [17]:

- Choose the projects, which are essential for short-, middle- and long-term success for the company

- Visualize dependencies between projects

- Regular coordination and tuning of the goals, milestones and costs of the projects in the portfolio

- Prioritizes project in times of resource shortages
- Communicate changes and troubles, occurring from changes in a project
- Identify conflicts between goals of different projects
- Develop scenarios to control crisis
- Systematically use synergies between projects
- Pass on knowledge from outcomes of a project
- Controlled cancellation of a project, which is not successful
- Complete and compare projects in an predefined way

There is a great variety of methods for analyzing these factors and as a result develop suitable product-strategies. The most commonly used methods are described in the following.

The BCG Growth-Share Matrix

The Boston Consulting Group (BCG) is a general management consulting company highly respected in business strategy consulting. The BCG Growth-Share Matrix is one of the many strategic concepts the organization has developed in the early 1970s and it is being taught at business schools and education programs around the world.

> ‘Assess strategically relevant business units, based on future benefit possibilities (market growth) and current market position (relative market share) [18]’

The BCG Growth-Share Matrix is based on two-dimensional variables:

- Relative market share
- Market growth

These two factors are often used as pointers to the healthiness of a business. That means, products with greater market share or within a fast growing market are expected to yield relatively greater profit margins. As seen in Figure 2.1 the x-axis defines the relative market share and the y-axis the market growth.

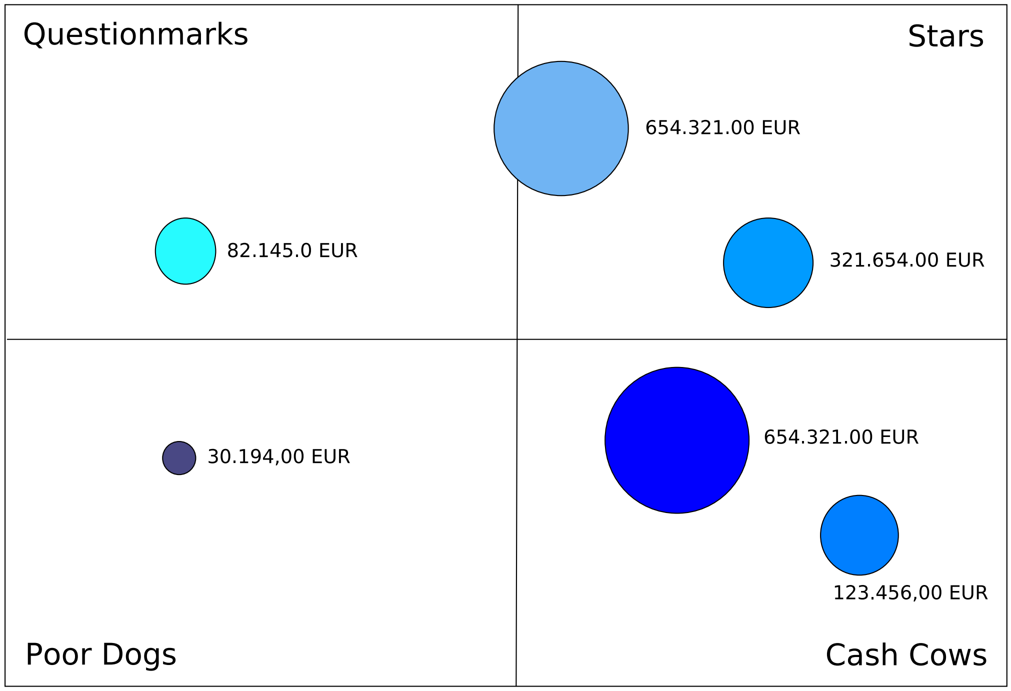

Figure 2.1: Boston Consulting Group Growth-Share Matrix [18]

- **Relative Market Share**
  The relative market share of a business unit or a product is captured with this parameter. It describes the cash generation - the higher the market share the more cash will be generated. To get the market share, a brand is compared to its largest competitor. Another way to drive down costs is the experience curve. The basic idea is, that the more often an organization performs a task, the better they get in doing this task, which results in lowering the operating costs. What this suggests is that the experience curve effect requires that market share is increased to drive down costs in the long run and at the same time a company with dominant market share will inevitably have a cost advantage over competing companies because they have a greater share of the market. Hence, market share is correlated to experience. (cp. [18])

- **Market Growth**
  The market growth axis, correlates with the product life cycle paradigm, and predicts the cash requirement a product has relative to the growth of that market. A fast growing market is generally considered attractive, and pulls a lot of organization's resources in an effort to increase gains. A case in point is the technological market widely considered by experts as a fast growing market, and tends to attract a lot of

competition. Therefore, a product life cycle and its associated market play a key role in decision-making.

The following passage describes the 4 quadrants of the BCG Growth-Share Matrix as seen in Figure 2.1.

- **Cash Cows**
  These products are said to have high profitability and require low investment due to the fact that they are market leaders in a low-growth market. The cash cows fund their own growth. They pay the corporate dividend, overhead, interest charges. They supply funds for research and development. In short - protect the cash cows.

- **Stars**
  Stars are leaders in high growth markets. They tend to/should generate large amounts of cash, but also use a lot of cash because of market growth.

- **Question Marks**
  Question Marks have not achieved a dominant market position and therefore do not generate much cash. They tend to use a lot of money because of growth of market conditions.

- **Poor Dogs**
  Poor dogs have only little future and are big cash drainers on the company as they generate nearly no money, caused by their low market share in a extremely low growth market.

The consideration of cash-flows of the individual business units is the fundament of value oriented management. Management should consider the investments not based on short-term revenue but with regard to long-term connection between the market and the competition. Also the product-lifecycle has to be considered.
The goal of each company should be a well-balanced product portfolio:

- Many business units in the cash-cow segment
- Several business units in the starting phase for the future
- As few business units as possible as question marks
- And nearly no business units in the poor dog segment

The BCG Portfolio Matrix combines the product lifecycle and the benefits of each product. The product lifecycle is explained in the next chapter.

SWOT Analysis

Another method to capture a company's position on the market is the Strength, Weaknesses, Opportunities and Threats Analysis (SWOT). It is a structured planning tool, which is used to measure the strategic position of a product or a product-group. For this reason the strengths and weaknesses in comparison to the biggest competitor regarding certain products on product-groups are evaluated and opportunities and threats are identified. [cp. [55]]

The basic form of the SWOT analysis was used as descriptive instrument. The gathering of specific strengths and weaknesses was captured via brainstorming. The delimitation of terms and the basic dynamic and complex connections were often not considered. As rough structuring- and classifying-grid four fields were used - strengths, weaknesses, opportunities and threats. During further development more attention was payed to the connection between strengths/weaknesses and opportunities/threats in the SWOT analysis. This was achieved through the consideration of trends and evolution on the market. The basis for this is the integration of a trend-analysis in the SWOT analysis can be seen in Figure 2.2. [cp. [55]]

- **Strength**
  These are the attributes of a company, which are useful to achieve the stated goal, e.g. dependability of a product, high productivity or level of awareness of the company.

- **Weaknesses**
  These are the attributes of a company, which may cause the company not to reach the goal, e.g. low marketing resources, small market share or weak level of support.

- **Opportunities**
  These are external, not influenced by the company, conditions, which helps the company to achieve the stated goal, e.g. growing market, untouched market potential in another country or a new available technology.

- **Threats**
  There are external conditions, which may cause the company not to reach the stated goal, e.g. increasing number of specialists, new legal restrictions or new products by competitors.

	Helpful in achieving the objective	**Harmful** in achieving the objective
Internal (attributes of the organization)	**Strengths**	**Weaknesses**
External (attributes of the environment)	**Opportunities**	**Threats**

Figure 2.2: SWOT Analysis

The first step in a SWOT analysis is the arrangement of all strengths and weaknesses in comparison to the products of the market leader. For this comparison, criteria such as product quality, lifetime or compatibility should be taken into account. They are listed in a matrix and assessed as seen in Figure 2.3 The second step is to identify product relevant trends in the market. After this step, the opportunities and threats are derived. During this step the evolution in view of the strength and weaknesses are evaluated. The fourth step is to prioritize the single opportunities and threats and derive measures and strategies for using the opportunities or cope with the threats.
Ideally a team which carries out the SWOT analysis, should consist of members of different departments, which represent a broad range of perspectives.

Products and product-markets are constantly evolving, to better understand these motions there are several models. The product life cycle model, which is described in the next chapter, is the most important of these.

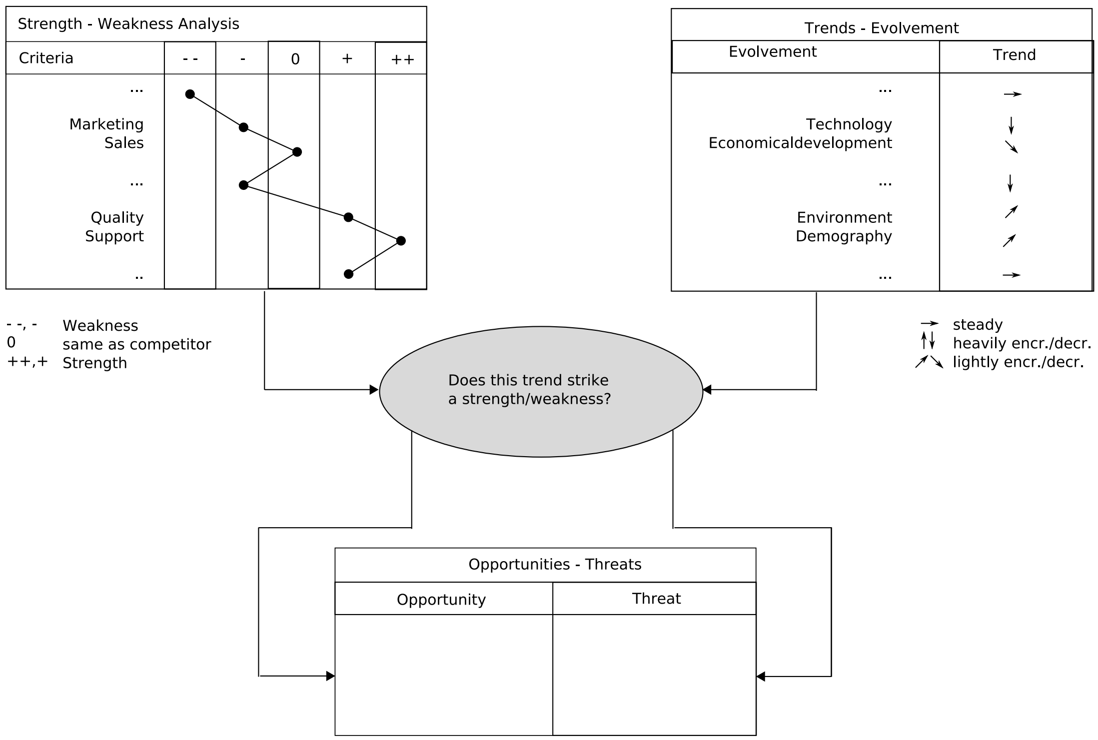

Figure 2.3: Strength - Weaknesses comparison ([55] page 247)

2.2 The Product Life Cycle

New products in general go through a sequence of different stages from development to introduction, growth, maturity and decline. This sequence is known as the product lifecycle. It represents the changes of the market, which influence the life of the product. The life cycle model can be applied to a brand or to a category of products. The duration may be as short as a few months for smaller products or up to a century or more for product categories, such as the automobile. The products may also vary but the underlying life cycles are similar.

The following Figure shows the main phases of a product lifecycle and the revenue the product generates during these stages. The development of the product is the first stage of the product life cycle. There are no sales but high expenses have to be made because of development. During this stage the company prepares for the introduction of the product. As the product progresses through its life cycle, changes to the product usually have to be made, to cope with the changes on the market. These changes are represented by the four major stages of the product life cycle: the introduction stage, the growth stage, the maturity stage and the decline stage. The following section will further describe these stages.

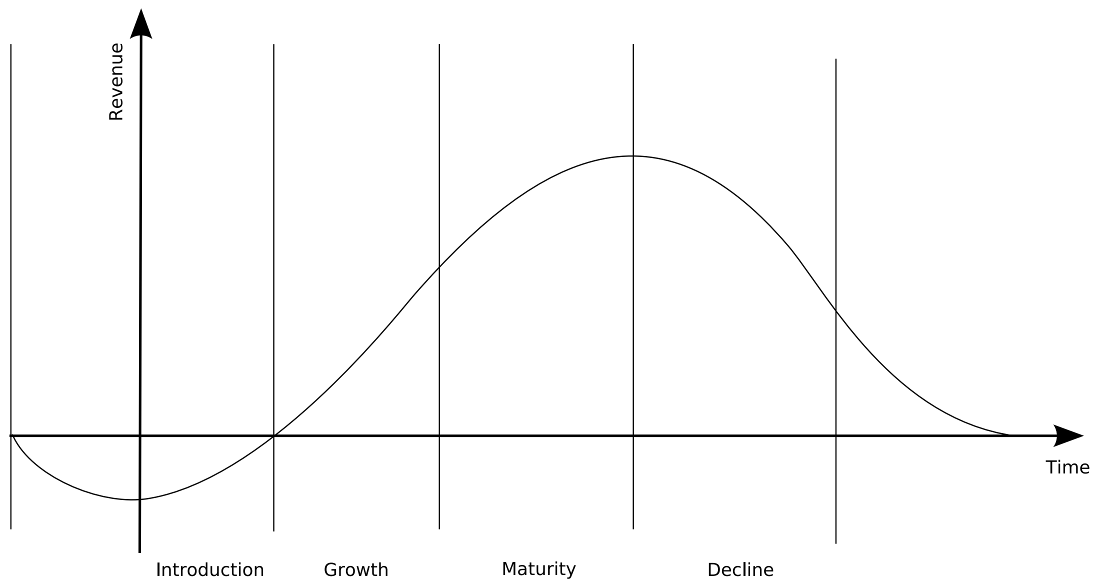

Figure 2.4: Division of the product life cycle into four sections [5]

1. **Introduction Stage**
   After the product has been introduced to the market, sales will be low until the customers are aware of the product and its benefits. Some companies are announcing their products before they are actually introduced, but such announcements also alert competitors and remove the element of surprise. A lot of money has to be spent on advertising and communicating the benefit of the new product, in order to increase the customer awareness of the product. During this stage, the company additionally has to cope with associated cost because of the initial distribution of the product. Because of these costs and the very low sales volume, companies usually make negative profits during this stage.

2. **Growth Stage**
   During this stage the revenue is growing rapidly. Sales are increasing as more customers become aware of the product and most importantly of its benefits. Additional market segments can also be targeted. Once the product has become a success and customers start asking for it, sales will increase further. Additional retailers will become interested in the product. The distribution of the product can be increased further. If a competitor enters the market, mostly at a later part of this stage, a price competition may be started and further funds for advertising have to be spent, to

convince the customer that the company's product is better than its competitors. To summarize, the main goal for this stage is to gain customer preference and increase sales.

3. **Maturity Stage**

   The maturity stage is the most profitable stage of the product life cycle. Sales continue to increase during this stage. Because brand awareness is strong, advertising expenditures will be reduced. The market share and maybe also the prices are decreasing because of competition. Also the competing products may be very similar, which results in difficulties of differentiating the product from the others. The company puts more effort into increasing the sales figure by encouraging competitors' customers to switch, increasing usage per customer, and converting non-users into customers. Retailers will also be encouraged to scale up sales and shelf space of the product, by granting special sales promotions to them.

4. **Decline Stage**

   The market is getting saturated and therefore the sales are declining. The product becomes technologically obsolete, as there are newer successors to the product. The customers may also be interested in new kinds of products. If the product has developed brand loyalty, profitability may be maintained longer. Unit costs may increase with declining production volumes and eventually no profit can be made. During the decline phase, the firm generally has three options:

   - Maintain the product in hopes that competitors will withdraw. Reduce costs and find new uses for the product.
   - Harvest it, reducing marketing support and coasting along until no more profit can be made.
   - Discontinue the product when no more profit can be made or there is a successor product.

The product life cycle is of course implemented in a product development process. The next chapter will present two models used in today's business world.

2.3 Product Development Process

A product development process (PDP) is in general a process used to create new products. Every stage of this process is clearly defined, from the initial phase until the cancellation of the product and knowledge gathering. This process is situated at the business level and used by managers to decide whether an idea or a project is worth developing or not. A PDP also incorporates and reflects the lifecycle of a product. Different to the development processes presented in the next chapter, a PDP is situated on the management layer of a company and deals with the complete lifecycle of the product. The development processes are located on project level of the company and only deal with the product during its development phase.

The following chapter describes two representatives of the product development process, ABB and prodSEM.

2.3.1 ABB Gate Model

The ABB Stage Gate Model defines eight gates where major business decisions are made. The basic layout of this model can be seen in Figure 2.5.

Definition of Gate Model Roles

There are two different roles defined in the ABB Gate Model - the Gate Owner and the Gate Assessor.

- The **Gate Owner** can be a person or a group of persons, who are responsible for the gate. They also have the authority to decide whether a product should be developed or not. The Gate Owner acts as the client, who commissions a project, therefore he is responsible for the funding and the availability of resources necessary for the project.
- The **Gate Assessor** is not included in the project. His task is to evaluate the project and the product before a gate, produce an assessment report and present a recommendation, for the decision, to the Gate Owner. The Gate Assessor has a lot of responsibility and therefore should be experienced and competent. [6]

Gate Procedure

The Gate Procedure consists of only two tasks - the Gate Assessment and the Gate Meeting. Input to the gate assessment is the document prepared by the Gate Assessor, interviews with project stakeholders and a gate assessment checklist. Gate assessment is done prior to the meeting over about a week and it is done by the Gate Assessor and the project manager. The results of this assessment are sent to all participants of the Gate Meeting, to prepare themselves. At the Gate Meeting a go/no-go/redo/hold decision for the project is made and this decision is communicated to all stakeholders after the meeting. [6]

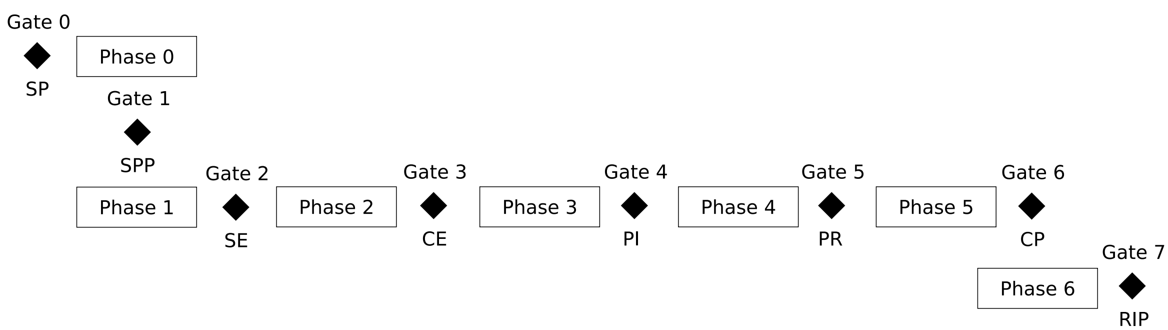

Figure 2.5: Overview of ABB Stage Gate Model [74]

<u>Gates</u>

These decision gates are introduced at [6]. There are altogether eight gates. The first six gates decide whether a project will be or will not be continued. The last two gates are used to close the project and gather all the “lessons learned”.

- Gate 0 - Start Project (SP)
  Input to this gate is a project proposal. Analysis of the market, competitors, intellectual property, risks, needed resources and product strategy should be included in this project proposal.

- Gate 1 - Start Project Planning (SPP)
  The scope for the development projects should be defined, such as functions, features and quality. Business constraints should also be defined, so they can be used for planning

- Gate 2 - Start Execution (SE)
  Project planning should be finished. Time and cost estimations, quality and risks management should be done.

- Gate 3 - Confirm Execution (CE)
  All the major risk of the projects should be identified and solutions should be proposed as well as for technical issues.

- Gate 4 - Product Introduction (PI)
  The product should be feature complete and ready for testing.

- Gate 5 - Product Release (PR)
  The product should be ready for the release to the market and/or customer.

- Gate 6 - Close Project (CP)
  The development project should be closed and the hand-over to manufacturing and/or service and maintenance should be confirmed.

- Gate 7 - Retrospective Investigation of Project (RIP)
  An evaluation of the project and product should be performed to evaluate the business success.

The approach used in the Stage Gate model is also used by several other product planning processes. Evaluating at each decision gate, whether the project should continue or whether the phase should be repeated ensures, that the products created during this phase are of high quality.
The following chapter will present a different product planning process, which is also used in today's business world.

2.3.2 Siemens prodSEM

prodSEM is a development model, which was created and is used by Siemens. This diploma thesis is using version 2.0, which was released in January 2005. The idea behind this model is similar to ABB's Stage Gate model. There are five phases and after each phase a go/no-go decision is made. A coarse overview is depicted in Figure 2.6 and the phases are described the following.

1. Initiation
   In accordance with the strategic placement of the company, new business ideas are evaluated. It is very important to assess the risks, performing for example SWOT analysis, and to evaluated the economy of the product.

2. Definition
   This phase is divided into three sub-phases: information gathering and rating, requirements analysis and product design. During this phases the product is defined in detail, to efficiently create the product during the next phase.

3. Realization
   The previously defined product is implemented during this phase. It is divided into four sub-phases: the implementation of a prototype and a pre-serial production sample, the preparation of the serial production and the serial distribution.

4. Operation
   This is the actual commercial use of the product. It is divided into two sub-phases: product support and product outlet.

5. Closure
   The products lifecycle is ended during this phase. Information gathered during the lifecycle of the product is processed and evaluated.

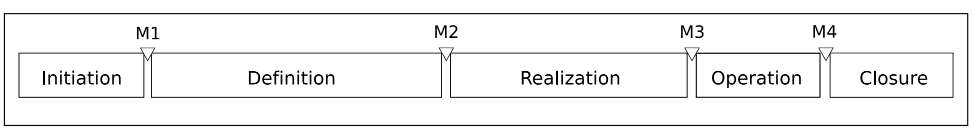

Figure 2.6: Overview of Siemens prodSEM [86]

The prodSEM is located at the business layer and therefore covers the business aspect of the product development process. Products defined in a product development process have to be implemented and developed using a development process for example hsSEM.

3 Engineering Processes

This section will give the reader an overview of a few models, which are currently in use in the target field of software and system development. At first the definition of a software process will be clarified. What is a software process? “A software process is a set of project phases, stages, methods, techniques, and practices that people employ to develop and maintain software and its associated artifacts (plans, documents, models, code, test cases, manuals, etc.). Not only do you need a software process, you need one that is proven to work in practice - a software process tailored to meet your exact needs.” [56]

3.1 V-Modell XT

Since February 2005, projects with the German government have to follow a certain protocol to ensure an acceptable quality-level. This protocol is documented in a framework called V-Modell XT. It is the improved and extended successor of the V-Modell 97. An overview of the V-Modell XT is provided in the following.

The V-Modell XT is designed as guidance for planning and executing development projects, taking into account the entire system life cycle. Not only are the results, which have to be reached during each of the phases of a project, defined, but actual approaches of how to obtain these required results are also described. Additionally the V-Modell XT also specifies the responsibilities of each participant. It describes in detail, ”who” has to do ”what” and ”when” within a project.
Other guiding principles, e.g. ISO standards, are presently in use, but they are less concrete than the V-Modell XT because they, e.g., do not specify a product template. These standardized, methodical guidelines permit a systematic execution even of complex and extensive projects. Thus, projects get more planable, traceable and lead to high-quality results with greater reliability, which is advantageous for acquirer and supplier. The V-Modell XT also regulates the required project cooperation between acquirer and supplier. The responsibilities of both sides are specified. Thus, the V-Modell XT standards are an

important basis for contracts between acquirer and supplier. In addition, the V-Modell XT improves the comparability of offers.
Small business enterprises also profit from the V-Modell XT. The V-Modell XT provides them with the possibility of falling back on standardized and proven templates for development and management processes. Thus, small business enterprises can automate their processes with reasonable effort and can thereby reliably achieve high-quality development results.
Thus, the V-Modell XT can be used as basis for contracting, as process guidance and as basis of communication. [13]

As already stated the V-Modell XT is the successor of the old V-Modell 97. The V-Modell 97, completed in the year 1997 and never changed since then, needed to be adapted. It did not reflect the current state-of-the-art technologies anymore. Also the "lessons learned" from the old V-Modell were collected and translated into improvements which then were merged into the new V-Modell XT. This diploma-thesis is using version 1.2 of the V-Modell XT.

The V-Modell XT is a guide for projects and the following listed objectives, as stated in [1]:

- Minimization of Project Risks
- Improvement and Guarantee of Quality
- Reduction of Total Cost over the Entire Project and System Life Cycle
- Improvement of Communication between all Stakeholders

The V-Modell Structure

The V-Modell XT describes in detail "who" has to do "what" and "when" within a project. The following Figure provides an overview of the structure of the V-Modell XT.

Project Types

The V-Modell can be applied to a great variety of project constellations but not all of the projects follow the same procedure. Depending on some characteristics these projects can be further classified and sub-categorized into project types. The most important project

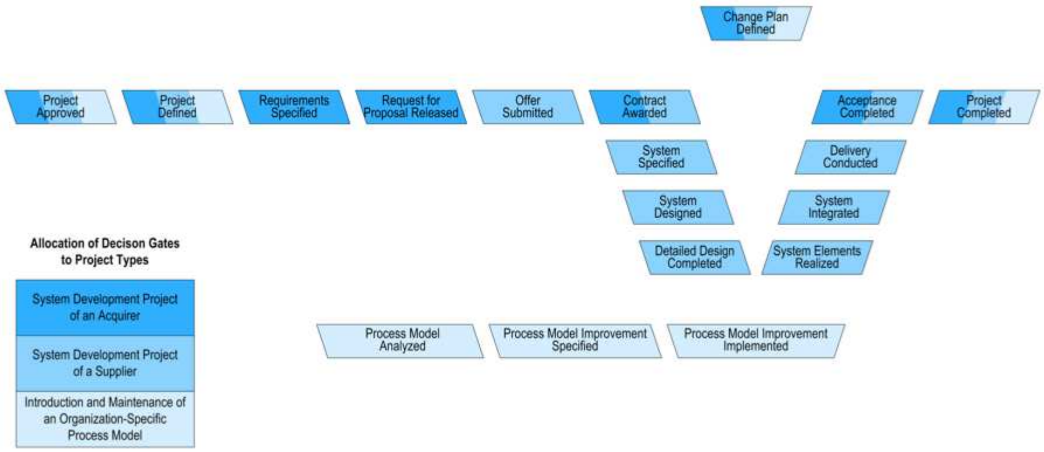

Figure 3.1: V-Modell XT Structure [1]

characteristics used for classifying the V-Modell projects are the subject of the project and the project role. The subject of the project of a V-Modell XT project can either be the development of a system or the development of an organization-specific process model. The project role can either be the acquirer or the supplier and designates the position of a V-Modell XT project in comparison to other projects. Each of the two project roles has its specific point of view of the project. As shown in Figure 3.2 the following project types

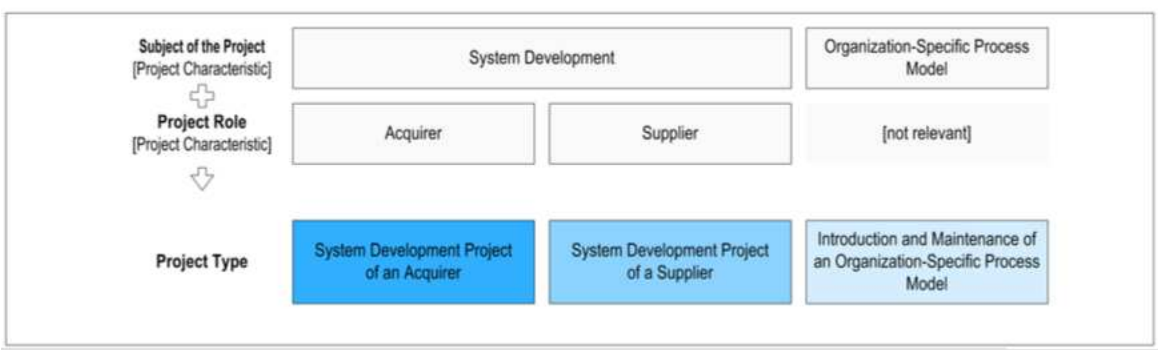

Figure 3.2: Classification of Projects and Subdivision into Project Types [1]

are specified based on the most important project characteristics:

- System Development Project of an Acquirer
- System Development Project of a Supplier
- Introduction and Maintenance of an Organization-Specific Process Model

The next definition described is the Process Module.

Process Module

A Process Module is a specific task, which has to be finished during a V-Modell XT project. Every process module is a self-contained unit and can be changed as needed. A typical process module, as shown in Figure 3.3, encapsulates the Work Product, Activities and Roles, which are necessary for the accomplishment of the process module. The V-

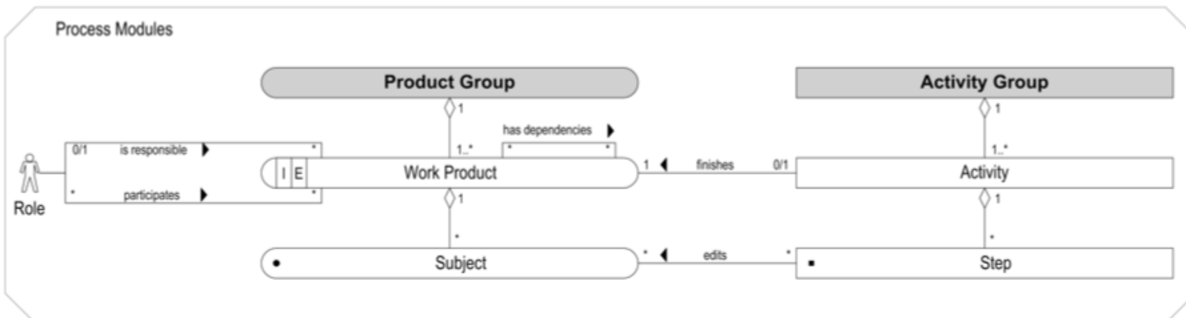

Figure 3.3: Process Modules and their Components [1]

Modell XT documentation includes for each role a detailed role description and for which products and activities the roles is responsible respectively collaborating. For a project it is important to group people together and assign roles to them, so that they can reach the project goals together. V-Modell XT defines a role as a organizational definition, which has abilities and knowledge, who gets certain tasks and permissions assigned. For every role there is a general definition of that role and a description of the tasks that role is responsible of respectively collaborating with. Also the qualification a person in this role needs to have. A complete list of roles defined can be found in the V-Modell XT documentation. [1]

Products in the V-Modell XT are structured hierarchically and are grouped into product groups. Product groups divide products according to their textual aspects. They are helpful to get an overview of all products in the V-Modell XT. There are 13 product groups defined in the V-Modell XT. The can further be divided into three scopes project management, development and organization. Also a complete list of products can be found in the V-Modell XT documentation. [1]

Combining all three - products, activities and roles - a process model covers “what” should be done in the actual project and “who” is responsible for the product. The question “when” a process model should be finished is answered by the “Project Execution Strategies”.

Project Execution Strategies

The sequence of a project with regard to contents and time is complex. The V-Modell XT provides a set of so-called "Project Execution Strategies". These project execution strategies define a basic framework for the ordered and replicable execution of a project. At least on suitable execution strategy for every "Project Type" is offered. The following Figure shows the available project execution strategies depending on the System Life Cycle Sections. The project execution strategies provide a basic set for the ordered and

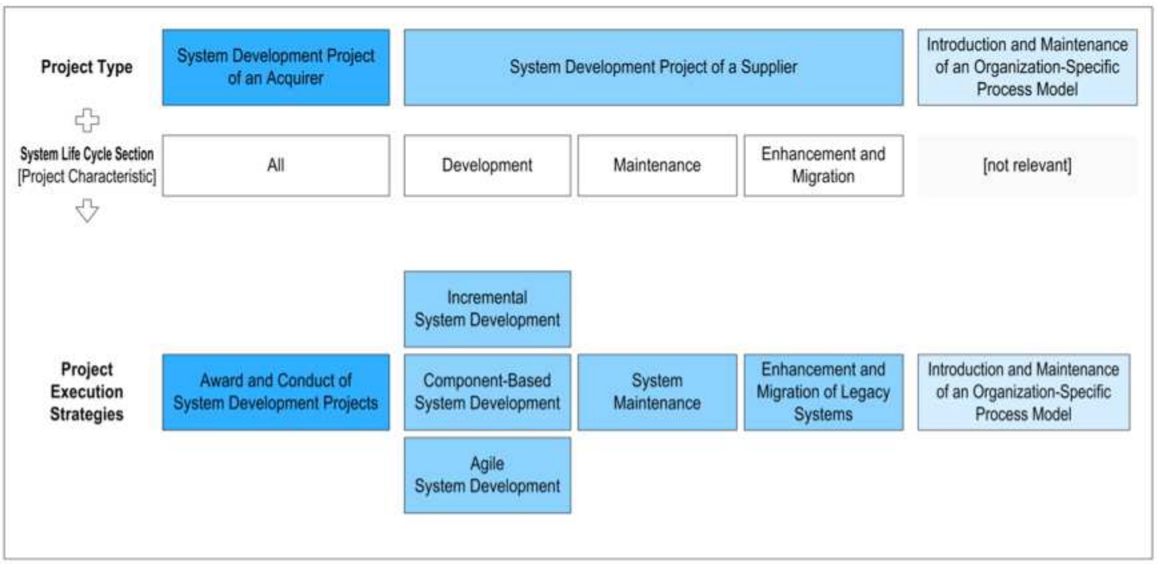

Figure 3.4: Project Execution Strategies to Project Types [1]

replicable execution of a project. Every project execution strategy specifies the sequence of the "Project Progress Stages", which have to be achieved during the project. A decision gate, shown in Figure 3.5, depicts these achievements.

Decision Gates

Figure 3.5: Project Execution Strategy, Decision Gates and Products [1]

The following Figure shows all decision gates in the V-Modell XT. The different coloring indicates in which Project Type and which project execution strategies the individual decision gates are used. Figure 3.6 shows all the decision gates of the V-Modell XT.

These decision gates can be tailored to fit several specific execution strategies. The three different colors represent the usage of the decision gates, which represent the different views on a project. The V-Modell XT covers the development of one product and is quite

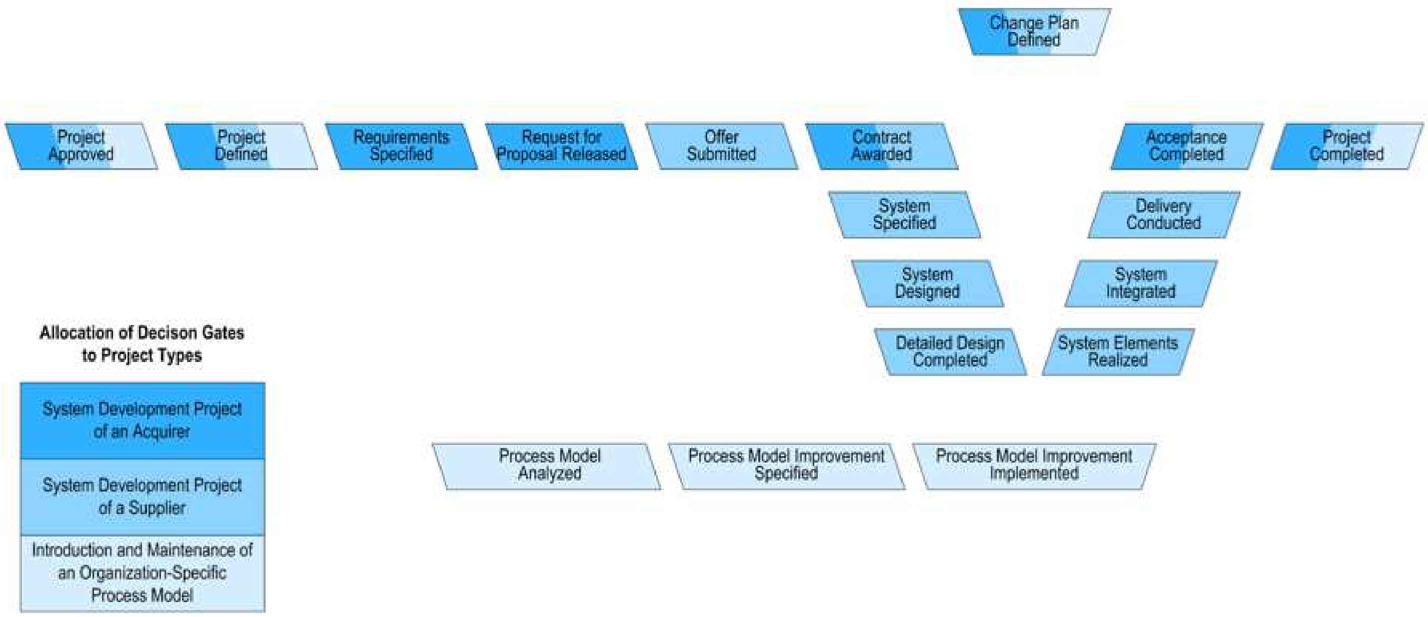

Figure 3.6: Decision Gates [1]

technical. Usually companies have several simultaneous projects in development. These projects are managed in a project portfolio, as described earlier.

3.2 Unified Process

The Unified Process (UP) is an iterative and incremental software development process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget. (cp. [28])
The basic concept is that a team works through appropriate workflows in an iterative manner, so that at the end of each iteration, an executable product is available. This reduces the risk of the project, by improving the communication between the team and the customer. The UP is organized in phases, as shown in Figure 3.7. These phases are not like the waterfall approach in the traditional sequence of requirement analysis, design, coding, integration and test. They are completely orthogonal to the traditional phases. Each phase in the UP is finalized with a milestone. The four phases are shortly described in the following paragraphs. (cp. [56])

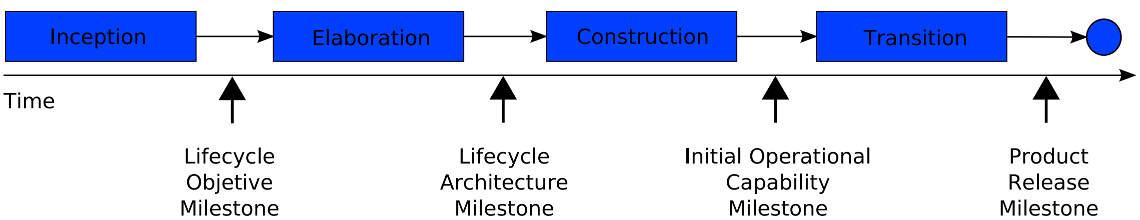

Figure 3.7: UP: The four phases and their milestones [28]

- Inception
  This phase starts with an idea, which provides a vision of the outcome of the software product. It specifies the business case and defines the scope of the project. At the end of the inception phase is the lifecycle objective (LCO) milestone.

- Elaboration
  Project management, planning the necessary work packages and resources is done in this phase. Specifying the features and the architecture of the product is also covered in this phase. The elaboration phase is concluded by the lifecycle architecture (LCA) milestone.

- Construction
  The construction phase is the phase where the product is developed, to meet the

initial vision of the product. This phase is concluded by the initial operational capability (IOC) milestone.

- Transition

  This phase is the final phase, where the product is delivered to the customer. This includes the delivery, training, support and maintenance of the product to satisfy the customer. This phase is finished by the product release (PR) milestone. This milestone is also the end of the cycle.

These four phases represent a development cycle and result in a software generation. A software product is created in an initial development cycle. Unless the life of the product ends at this point, the product will evolve into its next generation by a repetition of the sequence of inception, elaboration, construction and transition phases, but with different focus on the various phases.
The complete lifecycle is shown in Figure 3.8;

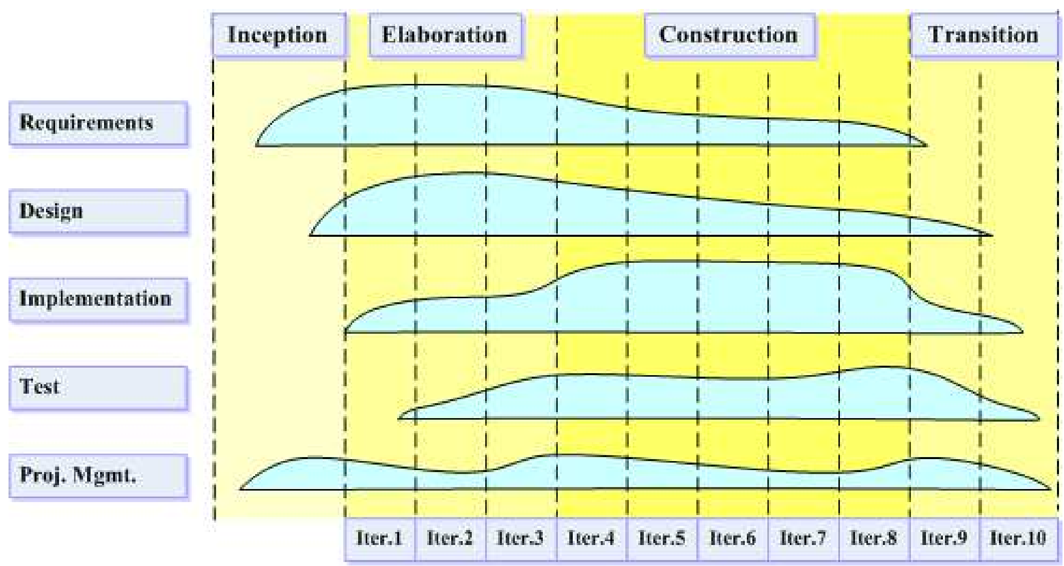

Figure 3.8: Unified Process [15]

Each phase has several iterations and each iteration results in an increment. Each increment is a release, which, compared to the pervious one, has new or improved features or stability. During each iteration, work has to be done in each of the disciplines - Requirements, Design, Implementation, Test and Project-management. Although the effort spent on these disciplines will shift during the course of the project.
In the Unified Process, use cases are used to gather all requirements for the product and

to define the work-package, that has to be done, during an iteration. Each iteration takes a few use cases and passes them through all the disciplines.
The development in the Unified Process is risk oriented. The greatest risks are addressed first.
The following describes the four phases in detail.

Inception Phase

During the inception phase the scope of the project and the business case are specified. The initial use cases for the product are identified and the most important ones are described. The primary goals of the inception phase are to achieve a common set of requirements between the stakeholder consensus regarding the objectives of the project and to obtain funding. To fulfill this task a high-level requirements model has to be developed to delimit the scope of the project. Also a high-level plan for how the project will proceed has to be developed.
The following goals have to be achieved to pass the Lifecycle Objective (LCO) milestone, which marks the end of the inception phase. [56]

- a consensus between the project stakeholders regarding the project's scope and resource requirements
- an initial understanding of the overall, high-level requirements of the system
- a justification or business case for the project, that includes economic, technological and operational issues
- a credible, rough schedule for the entire project
- a credible, detailed schedule for the initial iterations of the elaboration phase
- a credible, risk assessment and resource estimate/plan for the project
- a credible initial tailoring of the process software
- a comparison of the actual vs. the planned expenditures to date for the project and
- the development of an initial architectural prototype of the system [56]

Elaboration Phase

The primary goal of the elaboration phase is the detailed analysis of the problem domain and the establishment and validation of the system architecture. During this phase a detailed project plan, including cost and schedule estimates, based on the initial presumption from the inception phase, is developed for the subsequent construction phase. At this point the plan should be accurate and credible since it should based on the elaboration phase experience and since significant risk factors should have been addressed during the elaboration phase.

The architecture is validated primarily through the implementation of an Executable Architecture Baseline. This is a partial implementation of the system, which includes the core, most architecturally significant, components. It is built as a series of small, time-boxed iterations.

By the end of the Elaboration phase the system architecture must have stabilized and the executable architecture baseline must demonstrate that the architecture will support the key system functionality and exhibit the right behavior in terms of performance, scalability and cost.[56]

The Lifecycle Architecture Milestone marks the end of the Elaboration phase. At this milestone the decision is made whether to commit to the construction and transition phases. [12]:

- That the project vision is stabilized and realistic
- Upon requirements for the project
- The architecture is stable and sufficient to satisfy the requirements
- Risk management is continuing
- That the project has a realistic chance to succeed
- Upon detailed plans for the next phase, especially for the next few iterations

Construction Phase

Construction is the largest phase of the project. In this phase the remainder of the system is built on the foundation laid in Elaboration. System features are implemented in a series of short, time-boxed iterations. Each iteration results in an executable release of the software. The Initial Operational Capability Milestone marks the end of the Construction phase. [15]

Transition Phase

The final project phase is Transition. In this phase the system is deployed to the target users. Feedback received from an initial release (or initial releases) may result in further refinements to be incorporated over the course of several Transition phase iterations. The Transition phase also includes system conversions and user training. The Product Release Milestone marks the end of the Transition phase. [15]

Rational Unified Process

The Rational Unified Process (RUP) is probably the best known representative of the Unified Process. It was released in the initial version 5.0 by the Rational Corporation in December of 1998. Since 2002 the Rational Corporation is a division of IBM an therefore the Rational Unified Process is called IBM Rational Unified Process. In this diploma thesis this model is referred to as the Rational Unified Process (RUP).

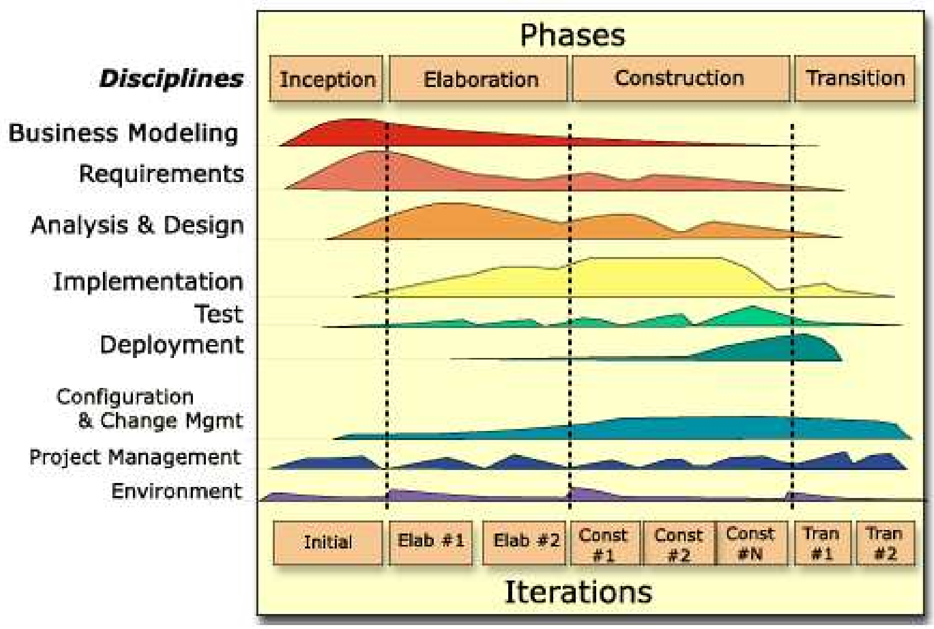

Figure 3.9: Rational Unified Process [16]

The RUP, as the Unified Process, is divided into four phases: the Inception, the Elaboration, the Construction and the Transition phase.
Each phase ends with a well-defined milestone. At these milestones, the continuation of the project is being assessed. The stakeholders have a look at the results of the particular increment, what has been done and what will be done in the next increments. At this point

a go/no-go decision is made, whether the project is to proceed to the next phase.
Within these phases, development is carried out stepwise , meaning that phases are further divided into iterations. An iteration is a distinct sequence of activities with an established plan and evaluation criteria, resulting in an executable release.
Figure 3.9 shows the two-dimensional structured model of the RUP. The vertical axis represents the different disciplines which are the general steps and the horizontal axis represents the time aspect in a project's life cycle. The vertical axis can also be seen as the representation of the static aspects of a project and the horizontal axis as the representation of the dynamic structure of a project.

The following passage describes the four phases of the RUP in detail.

Inception phase

In the Inception phase the business case, which includes business context, success factors, and financial forecast, is established. To complement the business case, a basic use case model, project plan, initial risk assessment and project description are generated. After these are completed, the project is checked against the following criteria:

- Stakeholder concurrence on scope definition and cost/schedule estimates.
- Requirements understanding as evidenced by the fidelity of the primary use cases.
- Credibility of the cost/schedule estimates, priorities, risks, and development process.
- Depth and breadth of any architectural prototype that was developed.
- Actual expenditures versus planned expenditures.

The milestone, called the Lifecycle Objective Milestone, marks the end of this phase. If the criteria are not met, the phase is repeated or the project is canceled.

Elaboration phase

The elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form. This phase must pass the Lifecycle Architecture Milestone by meeting the following criteria:

- A use-case model in which the use-cases and the actors have been identified and most of the use-case descriptions are developed. The use-case model should be 80% complete.

- A description of the software architecture to be used throughout the software system development process.
- An executable architecture that realizes architecturally significant use cases.
- Business case and risk list which are revised.
- A development plan for the overall project.

If the criteria for this milestone are not met, the phase has to be repeated or the project is canceled. After this phase, the cancellation of the project is much more difficult.

Construction phase

In this phase the main focus turns to the development of components and other features of the system being designed. This is the phase where the bulk of coding takes place. In larger projects, several construction iterations may be necessary in order to divide the use cases into manageable segments that produce demonstrable prototypes. This phase produces the first external release of the software. The Initial Operational Capability Milestone marks the end of the construction phase.

Transition phase

In the transition phase, the product has moved from the development organization to the end user. The activities of this phase include training of the end users and maintainers and beta testing of the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase. If it does not meet this level, or the standards of the end users, the entire cycle in this phase begins anew. If all objectives are met, the Product Release Milestone is reached and the development cycle ends.

RUP is based on a set of building blocks, or content elements, describing what is to be produced, the necessary skills required and the step-by-step explanation describing how specific development goals are achieved.

The main building blocks, or content elements, are the following:

- Roles (who)
  A Role defines a set of related skills, competencies, and responsibilities.
- Work Products (what)

A Work Product represents something resulting from a task, including all the documents and models produced while working through the process.

- Tasks (how)

  A Task describes a unit of work assigned to a Role that specifies a meaningful result.

The amount of disciplines is different to the UP. Within each iteration, the tasks are categorized into nine Disciplines, which are shortly described in the following.

<u>Business Modeling discipline</u>

The goal of the business modeling discipline is to understand the structure and the dynamics of the environment in which a system is to be deployed. This builds a bridge between system developers and users of the system so that both have a common understanding of the target environment. The second goal is to understand the problems in this organization and identify potential improvements. To achieve these goals, the business modeling discipline describes how to develop a vision of the new target organization and, based on this vision, to define the processes, roles and responsibilities of that organization in a model of the business. [28]

<u>Requirements discipline</u>

This discipline focuses on identifying all functional and non-functional requirements for the system and on establishing a vision and a better understanding of the system. Boundaries of the systems are also defined. Another important goal is to provide a basis for planning the technical content of iterations and to provide a basis for an estimate of the cost and time required to develop the system. [28]

<u>Analysis and Design discipline</u>

The goal of the Analysis and Design discipline is to transform the requirements, found in the previous discipline, into a specification, that can be used by system engineers to implement the system. Analysis focuses on ensuring that the system's functional requirements are handled. Design, on the other hand, handles the nonfunctional aspects, such as the implementation environment, performance and so forth. It focuses on the optimization of the system design. [28]

<u>Implementation discipline</u>

Implementation comprises transforming the chosen design into actual code and performing unit tests on the used components. The main goals of the Implementation discipline are the definition of architectural layout of the system. How are the subsystems organized? How are classes and objects implemented

- To define the organization of the code, in terms of implemented subsystems organized in layers.
- To implement classes and objects in terms of components (source files, binaries, executables, and others).
- To test the developed components as units.
- To integrate the results produced by individual implementers (or teams), into an executable system.

Systems are realized through implementation of components. The process describes how existing components are reused, or new components are implemented with well defined responsibilities, making the system easier to maintain, and increasing the possibilities of reuse. [28]

Test discipline

This discipline focuses on performing an evaluation of the overall quality level of the system. This includes the identification of defects, verifying the proper integration and interaction of all components of the software and verifying, that all requirements have been correctly implemented. [28]

Deployment discipline

The purpose of deployment is to successfully produce product releases, and deliver the software to its end users. It covers a wide range of activities including:

- Producing external releases of the software
- Packaging the software
- Distributing the software
- Installing the software
- Providing help and assistance to users

Although deployment activities are mostly centered around the transition phase, many of the activities need to be included in earlier phases to prepare for deployment at the end of the construction phase.The Deployment and Environment workflows of the Rational Unified Process contain less detail than other workflows.

Configuration and Change management discipline

The Change Management discipline in RUP deals with three specific areas:

- Configuration management
  Configuration management is responsible for a systematic structuring of the products. Artifacts such as documents and models need to be under version control and these changes must be visible. It also keeps track of dependencies between artifacts so all related articles are updated when changes are made.

- Change request management
  During the system development process many artifacts with several versions exist. CRM keeps track of all the proposals for change.

- Status and measurement management
  Change requests have states such as new, logged, approved, assigned and complete. A change request also has attributes such as root cause, or nature (like defect and enhancement), priority etc. These states and attributes are stored in a database so that useful reports the progress of the project can be produced. Rational also has a product to maintain change requests called ClearQuest. This activity has procedures to be followed.

Project management discipline

Project planning in the RUP occurs at two levels. There is a coarse-grained Phase plan which describes the entire project, and a series of fine-grained Iteration plans which describe the iterations. However, this discipline of the Rational Unified Process (RUP) does not attempt to cover all aspects of project management. For example, it does not cover issues such as:

- Managing people: hiring, training, coaching

- Managing budget: defining, allocating, and so forth

- Managing contracts, with suppliers and customers

This discipline focuses mainly on the important aspects of an iterative development process:

- Risk management
- Planning an iterative project, through the lifecycle and for a particular iteration
- Monitoring progress of an iterative project, metrics

The project management discipline contains a number of other Plans and Artifacts that are used to control the project and monitoring its performance such Plans are:

Environment discipline

The environment discipline focuses on the activities necessary to configure the processes for a project. It describes the activities required to develop the guidelines to support a project. The purpose of the environment activities is to provide the software development organization with a software development environment, both processes and tools, that will support the development team. The Environment discipline workflow is broken down into three main steps: Prepare Environment for Project. Preparing the development environment for a means turning the underlying development process into an enactable project-specific development process. This involves :

- defining how the project is going to use the configured development process.
- establishing a development case describing deviations from the underlying process.
- qualifying artifact selections with timing and formal requirements.
- preparing project-specific assets, such as guidelines and templates, according to the development case.
- producing a list of candidate tools to use for development.

Prepare Environment for an Iteration. The purpose of this workflow detail is to ensure that the project environment is ready for the upcoming iteration. This includes process and tools. This work is focused mainly on:

- Complete the Development Case to get ready for the iteration.
- Prepare and, if necessary, customize tools to use within the iteration.
- Verify that the tools have been correctly configured and installed.

- Prepare a set of project-specific templates and guidelines to support the development of project artifacts in the iteration.
- Make sure that all the changes made to the project environment are properly communicated to all the project members

Support Environment during an iteration. Support the developers in their use of tools and process during an iteration. This includes installation of required software, ensuring that the hardware is functioning properly and that potential network issues are resolved without delays.

3.3 Agile Development Methods

Before several of the current agile development methods are described, the history and the origin of agile development is presented shortly.
As a reaction to the software crisis in the 1960ies, the term software engineering was created. To date several development methods have been created. The drawback of those developed methods is their heavy weight and their missing rigidity. Another factor, which is still present until the current day, is that software development is connected with a negative image. Software projects are often not finished on time or do not show the required results.
Originating from the dissatisfaction of current software development methods, an agile approach was created. There are four values and twelve principles of the agile development approach, which are written down in the agile manifesto [36]. The four values are: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan. The twelve principles and further information about the Agile Manifesto can be looked up at [36].

3.3.1 Scrum

The term “scrum” takes its origin in the sport Rugby. It describes the part of the game, where both teams wrestle for the ball. Ken Schwaber and Jeff Sutherland presented Scrum at the OOPSLA‘95 for the first time. Both were searching for better development-methods for their company. The focus of Scrum is the management of teams, which are, not necessarily developing software, but in general producing any kind of product. Therefore Scrum has no explicit practices for design, development or testing of software. Scrum can be used for any kind of project, where one or several teams are working together. [19] Scrum has two main goals: adaptiveness and team empowerment. The management should support the teams and keep them away from barriers, not bother them with unuseful discussions and talks. A great deal of responsibilities is handed over to the team, but this implies enough trust into the team. Therefore Scrum demands that team members follow the following principles: [40]:

- Commitment:
  Be prepared to commit yourself for a certain goal.

- Focus:
  Finish your work and focus on your todos. Anything else does not have to bother you.

- Frankness:
  In Scrum, everything is visible for everyone.

- Respect:
  It is most important so respect your team-members.

- Courage:
  Have the guts to commit yourself, to act, to be honest and to get respect.

Roles

Scrum defines five different roles: the Scrum Master, the Scrum Team, the Product Owner, the Customer and the Higher Management. The following describes the five roles in detail. [19]

- **Scrum Team**
  The main task of the Scrum team is the development of the software product. The teams consists of a maximum of eight people, who organize themselves and make decisions independently. The ideal Scrum team should be from a lot of different disciplines, to get many different points of view on the project.

- **Higher Management**
  Higher management has the final authority to decide and to give the project rough goals and requirements. The high management is also responsible for dealing with problems regarding the project environment, e.g. if the customer refuses to cooperate.

- **Scrum Master**
  The Scrum master is the interface between the Scrum team, the customer and higher management. His role is to keep the team from getting distracted in any way. It is his responsibility, that the principles of Scrum are being followed and that the project show the best possible progress. The Scrum master has no authority over the team, because the team organizes itself. He only interferes if the team is not able to make decisions. Therefore the role of Scrum master is better described as coaching than managing.

- **Product Owner**

  The product owner is responsible for the product “Backlog”. The blacklog is some kind of to-do list. The product owner assigns the tasks of the product blacklog, together with the team, to the single iterations (sprints) and he validates their implementation.

- **Customer**

  The customer is available for all other tasks, which are needed to complete the project successfully.

<u>The Process</u>

Scrum identifies three different phases: the Pre-Game phase, the Game phase and the Post-Game phase. The phases can be seen in Figure 3.10.

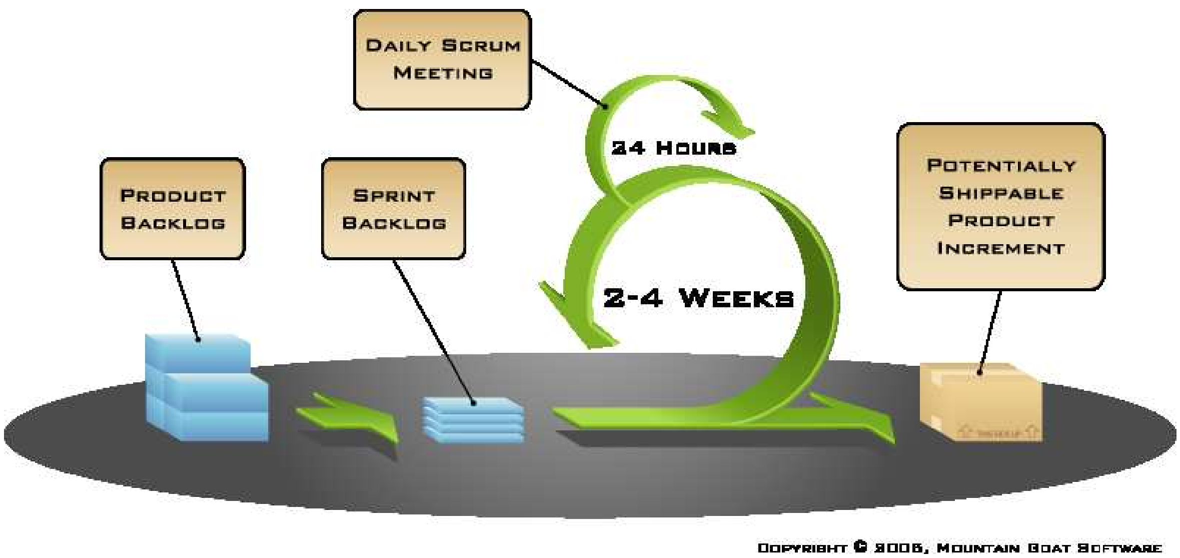

Figure 3.10: Scrum Phases: Figure taken from [84]

- **Pre-Game phase**

  The pre-game phase mainly consists of planning the project and designing the basic architecture. During project-planning on one hand the guidelines for the project are defined, e.g. who is be part of the project, which programming language or which developing tools will be used. On the other hand, the first version of the product backlog is created, containing all functionality, which the product has to include. After the creation of the product backlog, the basic architecture is designed, to get an impression in which direction the product should be heading. These design proposals are audited during a *design review meeting*. At the end a binding decision, which architecture are used, is made. In Addition a first release planning is done. This planning shows which features are available at which release.

- **Game phase**
  After the pre-game phase is the game phase, where the product will be created in iterative steps, that are also called *sprints*. Before a sprint starts, a *sprint planning meeting* is held. During this meeting the goal of the following sprint, the *sprint goal*, and the tasks to reach the goal are agreed upon. The result of this meeting is the *sprint goal* including the *sprint backlog*, which includes all activities, which have to be done, during this sprint. After the definition of the *sprint backlog*, the team can decide on its own, how the tasks are finished. Because of the changing requirements and information during this phase, scrum has no guidelines for the development of the tasks. Scrum only controls a few selected variables, which are characteristic for the development process. An important part of ongoing control is the daily *stand-up-meeting*, where each team member shortly presents his/her work and informs the scrum master of any disturbances, so the scrum master can get rid of those disturbances. As soon as a sprint finishes, that is when all tasks in the sprint backlog have been completed or a maximum of 30 days has passed, a *sprint review meeting* is scheduled and the decision for a follow up sprint is made. If a follow up sprint is approved, the tasks for this new sprint are defined. Otherwise, this phase is concluded and the project moves on to the following phase - the post-game phase.

- **Post-Game phase**
  The developed software should be in a runnable state after three to eight sprints. Before the product is shipped to the customer, it is tested in a realistic environment. If no errors are found, the software is released and documented. Should errors be encountered. they are incorporated into the backlog of a new sprint, that subsequently deals with them.

3.3.2 Agile Unified Process

The Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP). It is based on the RUP but uses an simpler approach using agile development techniques.

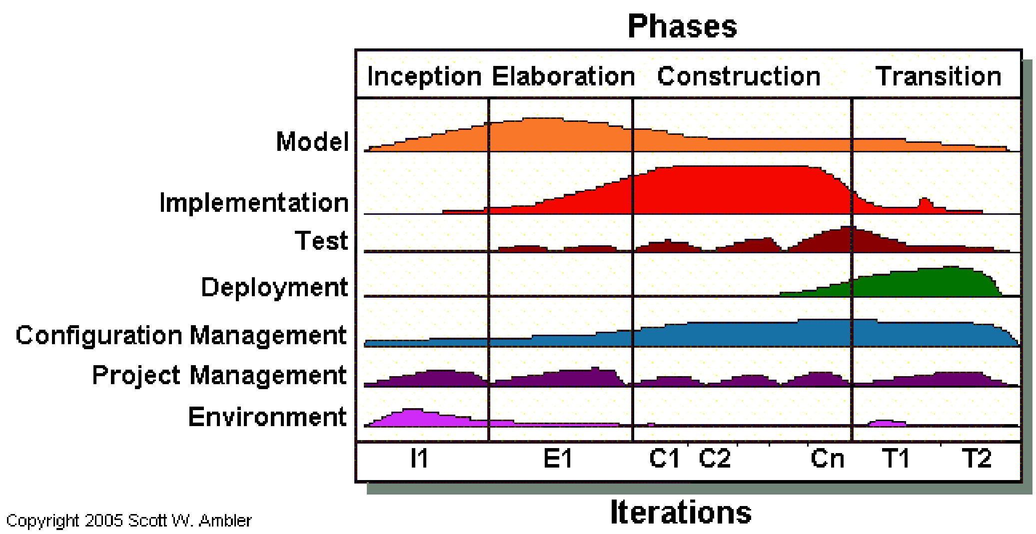

Figure 3.11: Agile Unified Process [21]

AUP is based on the following principles of agile techniques [21]:

- The staff knows what they are doing
  The development team will not read detailed process documentation, but they do need some high-level guidance.
- Simplicity
  Every aspect of the work is described incisively and understandable, not using thousands of words.
- Agility
  The Agile UP follows the values and principles of the Agile Alliance.
- Focus on high-value activities
  The development team should focus on activities, which actually count, not on every possible aspect, which possibly can happen to the project.

- Tool independence
  Any toolset can be used with the AUP. It would be best to use tools, that are best suited for the job.
- The AUP has to be tailored to meet your own needs
  The AUP product, a HTML version of the AUP process, can easily be tailored using any HTML editor.

As seen in Figure 3.11, the AUP, like UP and RUP, has four phases, but there are only seven disciplines. The following explains the seven disciplines in more detail.

Model
The goal of this discipline is to understand the business of the organization, the problem domain being addressed by the project, and to identify a viable solution to address the problem domain.

Implementation
The goal of this discipline is to transform your model(s) into executable code and to perform a basic level of testing, in particular unit testing.

Test
The goal of this discipline is to perform an objective evaluation to ensure quality. This includes finding defects, validating that the system works as designed, and verifying that the requirements are met.

Deployment
The goal of this discipline is to plan for the delivery of the system to end-users.

Configuration Management
The goal of this discipline is to manage access to your project artifacts. This includes not only tracking artifact versions over time but also controlling and managing changes to them.

Project Management
The goal of this discipline is to direct the activities that take place on the project. This includes managing risks, directing people (assigning tasks, tracking progress, etc.), and coordinating with people and systems outside the scope of the project to be sure that it is

delivered on time and within budget.

Environment

The goal of this discipline is to support the whole effort by ensuring that the proper processes, guidance (standards and guidelines), and tools (hardware, software, etc.) are available for the team as needed.

3.3.3 Extreme Programming

In the early 1990's Kent Beck, Ward Cunningham and Ron Jeffries introduced the Extreme Programming (XP) method. XP targets small to medium sized teams of developers, who develop software with vague or rapidly changing requirements. The basic assumption and criticism to traditional development methods is, that developers using XP can avoid the high costs of change using technologies such as objects, patterns or relational databases. [3]

The following describes XP, its values and its methods. The four main values of XP are [24]: Communication, Simplicity, Feedback and Courage.

Communication

Projects often do not succeed because of a lack of communication. Either information is not passed on or a neglect of the received information, e.g. programmers do not need the requests made by customers. XP tries to keep communication intact and therefore provides many practices, which cannot be done without communicating with each other. XP uses practices such as unit testing, pair programming and task estimation. Using these practices, programmers, customers and managers have to communicate with each other. [24]

Simplicity

XP states, that it is better to do a simple thing today and pay a little more tomorrow to change it if it needs to be changed. The idea is, not to implement to best or the most complex code, rather than code that works and fulfills the requirements. In an environment of rapidly changing requirements, it is impossible to know if a piece of code, that took hours to implement, is really included in the final product.

Feedback

Besides communication and simplicity, feedback is another very important value in XP.

Using unit tests and functional tests, programmers get immediate feedback whether their code works or not. Again increases of communication because of feedback. Errors or wrong assumptions can only be avoided through quick feedback.

Courage

Last but not least, XP expects courage from its development team. It expects courage to explore new paths. For example the disposal of a day's work can often be more useful, than clinging to a certain solution.

The XP lifecycle encompasses four basic activities, which can be seen in Figure 3.12: Planning, Coding, Testing and Designing

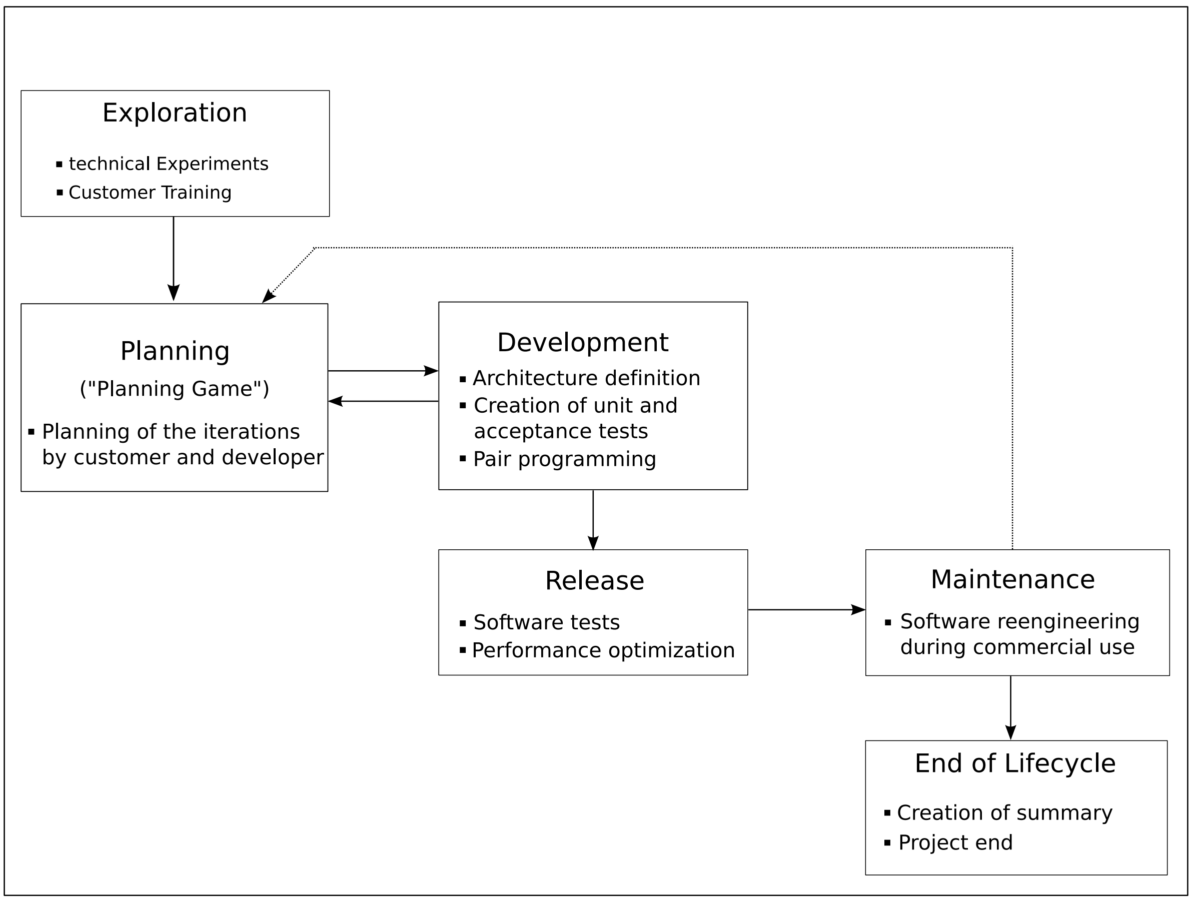

Figure 3.12: XP Processmodel [19]

XP also consists of four important values:

- Continuing communication with the customer
- Simplicity, which is achieved by a constant focus on minimalist solutions
- Immediate feedback through unit and functional testing
- The courage to act and address problems actively

These values are not completely new. Values like communication, simplicity or user involvement are part of any disciplined process. But the XP takes these values to the 'extreme', see Table 1. Simplicity in term of XP does not mean, that the developers are

Commonsense	XP extreme	XP implementation practice
Code reviews	Review code at all times	Pair programming
Testing	Test at all times	Unit testing, function testing
Design	Make design part of everybody's daily business	Refactoring
Simplicity	Always work with the simplest design that supports the system's current functionality	The simplest thing that could possibly work
Architecture	Everybody works to refine the architecture at all times	Metaphor
Integration testing	Integrate and test several times a day	Continuous integration
Short iterations	Make iterations extremely short - seconds, minutes and hours rather than weeks, months or years	Planning game

Table 3.1: The Extreme Programming way [2]

hacking code, it rather means focusing on the part with the highest priority or the part which is, at this point in time, the most important. Various XP developers may be using different XP practices, but they consist of twelve basic elements:

- Planning game: New version of the software are planned. The features and the effort are estimated.
- Small releases: Problems, which normaly only occur at big releases are identified earlier because of small releases.

- Metaphor: Choose a metaphor, which is understood by customer and development team, to overcome the barrier of technical term.
- Simple design: The simpliest solution, which solves a problem, should be used.
- Testing: There should be an automated test for every piece of code. No code shall be written without the proper automated test.
- Refactoring: Constant achritecture, desing and source code refactoring and improvement has to be done.
- Pair programming: The developers are divided into pairs - one developer is coding, the other one has the big picture in mind. The roles are changed regularly.
- Collective ownership: The whole team is responsible for the code. Only the team can be successful.
- Continuous integration: To give the customer continous working releases and get immediate feedback.
- 40-hour weeks: Overtime has to be avoided, because the motivation of the developers will decrease.
- on-site customer: Close cooperation with the customer to always know, what to concentrate on.
- coding standards: The team should follow common programming standard, which they have agreed upon.

A pool of 'stories' characterizes the systems full functionality. These stories are written by the customer and represent the functionality the system has to have from their point of view. During release planning a subset of these stories is selected, which then describe the work for the developer for the next release. The customer can add new stories at any time. Therefore the requirements of a product are very volatile and can change quite quickly. This situation is handled by a two-weeks increment, which means, that every two weeks new stories are agreed upon, that are to be implemented. XP developers generate a metaphor to provide the project's overall vision. Documentation seems to be left behind when using XP. Because XP encourages a continuing improvement and redesign. Only little attention is paid to detailed documentation of the design. [2]

Another method XP uses is pair programming. This method is quite controversial, mainly because of its resource consequences. It appears as if pair programming consumes twice a many resources, but on the upside leads to fewer defects and decreased cycle times. [4]

XP emphasis collective ownership, which means, that anyone at any time can change any piece of code. [2]

3.4 Comparison of the presented Engineering Processes

The following compares the presented models, which will be used to create the extension of the V-Modell XT, regarding their advantages and disadvantages. Obviously the extension should include as many, even perhaps, all the positive aspects of the presented models and as little drawbacks as possible. This is exactly what the extension to the V-Modell XT is focusing on - it uses the pros and avoids the cons.

Process	Area	Advantages	Disadvantages
Scrum	Software Development	+ Great emphasis on team work + Team learns and contributes throughout the process + Team becomes autonomous and strives for excellence + Can act as a wrapper for practices already in place	- The team has to be committed to the project - The size of the team is restricted - Only suited for development of new products - Relies on experience
XP	Software Development	+ XP encourages people to increase communication + It builds competence in all team members + It provides management with several additional tools	- Duplication has to be eliminated at all costs - Tests have to be written for everything. 'Do not write code unless there is a test for it.' - The temptation to let iteration deadlines slip is high
RUP	Software Development	+ Risks are taken care of earlier + Change is manageable + Opportunity of reuse	- Needs customization before it can be used - It fails to provide any clear implementation guidelines
V-Modell XT	Software Development	+ Covers a lot of project management theory + Already has predefined deliverables + Assures project stability	- Complex and huge model - Requires a lot of work for adaptation - No processes for concurring business planning
ABB's Stage Gate	Product Management	+ Several points where success of a project is assessed + Guarantees quality of the project	- Not a development model, but rather a guideline - Has to be combined with another model

Table 3.2: Summary of the presented models

4 Research Approach

This chapter motivates and outlines the research goal of this diploma thesis. The main focus of this diploma thesis is, how management and engineering, and therefore product planning and product development can work together using only one process model, the V-Modell XT. Afterwards the evaluation of the developed extension, including the hypotheses, is defined and explained.

4.1 Research Goal

The field of software engineering chronically suffers from delayed products and, in addition to this deficit, an increasing dissatisfaction of customers, because they do not get what they really wanted. Product creation projects are often not successfully finished because of misunderstandings and missing respectively too little communication between management and engineering. This situation is very prominent and occurs especially during product development, where management and engineering should work together intensively, but do not.

Another currently investigated problem concerns the need for portfolio management in small companies. Do small companies really need portfolio management? Vähäniitty comes to the conclusion [83], that also small software development companies, which not only supply software products but also offer services, should also make use of portfolio management to most efficiently distribute their development resources and thereby saving money by improving efficiency.

Arising from this condition, this diploma thesis focuses on two research questions:

1. ‘Is it possible to create one process model, which can be used by management and by engineering, to successfully create products?’

2. ‘Can this process model also be used by small companies, to develop their products?’

This first research question deals with the cooperation of management and engineering within one company. By including management and engineering in one process during the lifecycle of a product, the quality and the possibility that the product is finished on time and within budget increases. Most of the time management, which performs market and competitor analysis, comes up with the idea of a new product. They have a vision, what the product should look like and which features should be included. Engineering on the other hand does not have this big picture concerning the product and therefore has no idea on which feature they should concentrate on respectively which part of the product they should implement most efficiently. Management and engineering have to work together and the final goal must be well known throughout the company, to give everybody the opportunity to contribute to the result to their best of their abilities. This can be accomplished by increasing the cooperation of management and engineering. This diploma thesis not only investigates this very issue but it also suggests a possible approach to this issue.
The second research question focuses on the application of the new extension to the V-Modell XT and of course the application of the V-Modell XT to the processes of small companies. Most often these companies only consist of a few people and, while already performing their work as good as possible, these companies often have to struggle to meet deadlines of projects and to fund the resources for further projects. It is very common for small companies arise from a great new idea for a new product but with no funding. The founders of these companies invest all of their money in the foundation of the company and the development of this very product. Small software companies often lack proper resource planning and therefore spend their money suboptimally. They may have some kind of project management, but this project management is often limited to todo lists and some initial time planning. Resource planning therefore is crucial to small software companies, due to their limited resources. The V-Modell XT and the created extension require some effort to be implemented properly. This effort is very critical in small software companies, because of the described situation. The benefit of the implementation of the V-Modell XT and its extension is the proper management of new respectively existing products and a guideline for successfully managing the creation of new products, on which companies inexperienced with product management also can rely on.

Derived from the research questions, the goal of this diploma thesis is to develop a process model, which encourages the collaboration and communication between management and engineering. This can only be accomplished if both disciplines are using a model, which encourages collaboration. Therefore an extension to an existing process model, the V-Modell XT, is developed. This extension covers the product planning process, which, in combination with the V-Modell XT, gives companies a new tool, to successfully develop their products on time.

The basic idea behind the research goal, is the combination of several models from the management and engineering layer as shown in Figure 4.1. It can be seen, that the V-Modell XT already covers the whole software development part, but the parts before and afterwards are not integrated into the V-Modell XT. Therefore the analysis heading the whole product creation process has to be done by management. During this phase for example the feasibility and the revenue of a product idea are investigated and assessed. This assessment is done by using portfolio management and models such as the BCG matrix or SWOT analysis, as described in Chapter 2. Once the product is specified the engineering

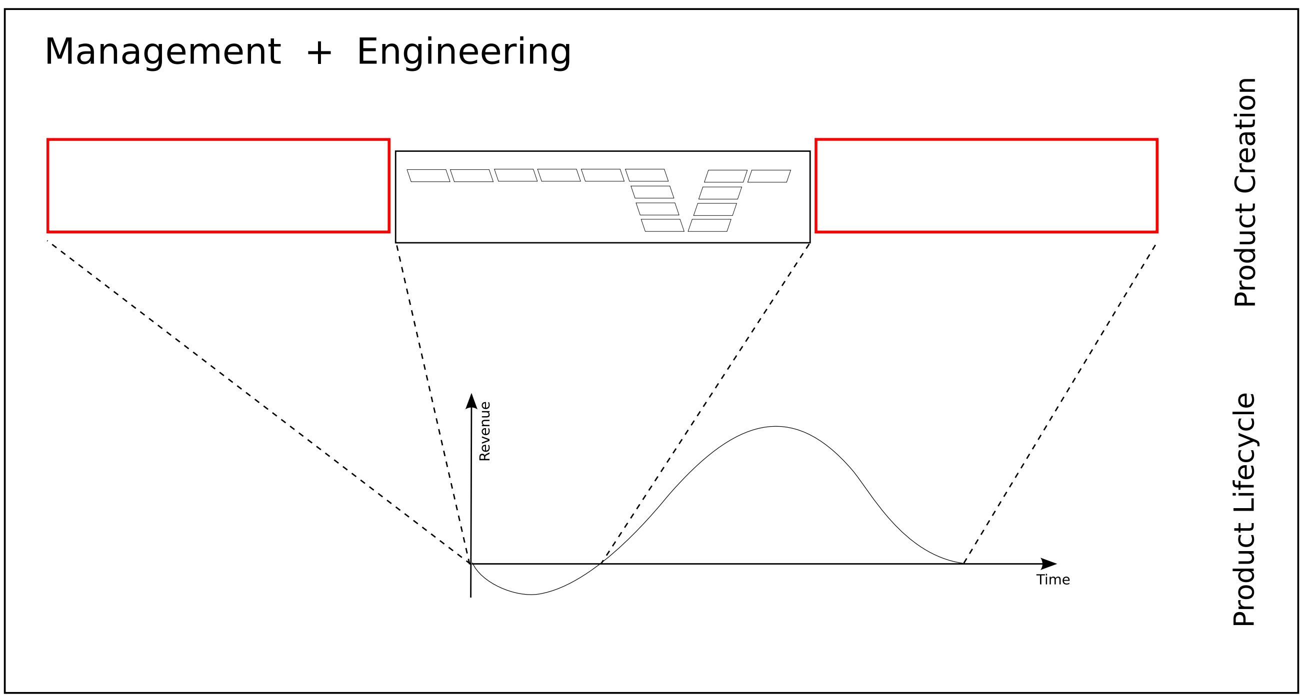

Figure 4.1: The basic idea of the research goal.

layer, represented by the V-Modell XT, takes over to implement this new product. After completion the actual life of the product, the commercial use, begins, which lasts until the decline phase, which ends the lifecycle of the product. This phase is again not covered by the V-Modell XT and therefore has to be added explicitly.

4.2 Solution approach

The first step to creating this extension is the definition of a meta-model, which covers the basic product planning process. The meta-model is the basis for the extension, which, in an abstract way, covers all phases and milestones during a product creation process. This meta-model is derived from existing concepts and models currently in use in the business and engineering world. It is based on models such as the product creation process, ABB's Stage Gate model and Siemens prodSEM. All these models and processes are described in Chapter 2.3. The meta-model additionally unites positive approaches from other models, as illustrated in Table 3.2.

During the next step both models, the new meta-model and the V-Modell XT, are broken down into small phases and the roles, activities and products of each phase are then compared to each other to determine their equality or to point out differences. The parts that are missing in the V-Modell XT but required by the meta-model, the initiation phase including product planning for example, are added using the BCG Matrix and SWOT analysis. On the development level, the V-Modell XT already has guidelines to use an agile development approach. Because only predefined features are developed during each iteration and progress is evaluated and measured repeatedly, XP is chosen, as a development method. By using this development model, management has an overview on what is done during an iteration and can therefore always keep track of the progress of the project. Changing requirements, during the course of a project can also be absorbed, in order not to endanger the outcome of the project. As already seen in Figure 4.1 the lifecycle of the product is not covered by the V-Modell XT and therefore has to be added. This addition can be accomplished by using a special project execution strategy included in the V-Modell XT. This execution strategy covers the maintainance and servicing of systems and it can be used to accompany the product throughout the lifecycle. New features, updates or change requests can be integrated during this execution strategy. Another benefit of using this approach is that it is already included in the V-Modell XT and therefore the roles, activities and products are well known and can now be used throughout the whole product lifecycle.

The result of this process is an extension of the V-Modell XT, which also includes the product planning procedures. It is presented in the following chapters.

4.3 Evaluation

The developed model is afterwards evaluated based on a product development project. The goal of this project is the development of a new product for a small company using the newly created extension of the V-Modell XT. This company is using an agile process to develop their products, therefore the special agile development strategy within the V-Modell XT using XP can be used, as described above. The evaluation will be measured using interviews with the project members, to get a qualitative estimation of the case study. This estimation shows the author if the proposed extension to the V-Modell XT is feasible. Quantitative estimations cannot be made, because of the missing comparison opportunity to similar projects of the project team.

Goal Definition

The case study is motivated by the research questions if the application of the extension of the V-Modell XT is feasible and if it is possible to apply the extension to the processes of a small company. The company in the case study is using Extreme Programming as a development method and up to now there was little focus on software quality and project documentation. Therefore the traceability of the projects was very poor and very strongly depending on the person, who implemented the program. Now that the first product is going to be released, it has to be ensured, that a certain level of quality is abided, not only in project documentation but also in development and management processes. For this reason the V-Modell XT and the created extension are introduced to this company during the case study and the results on their process and the outcome of the project is measured.

Object of study: The object of study is the newly created extension of the V-Modell XT and the ability to integrate this process framework into existing processes in a software development company.

Purpose: The purpose of the case study is to evaluate the feasibility of applying the extension to a project at all. The second question is, if the extension is applicable, can it also be used by a small development team or a small company?

Perspective: The perspective is from the researcher's point of view. The researcher would like to know if it is possible to use the extension in an effective way. Not only in a large company, but also in a small company, with only a few employees.

Quality focus: The main effect observed in this case study is the feasibility of the extension when used by a small team. The researcher focuses on the functionality planned and the functionality actually implemented - the degree to which the software is finished. The quality of documentation or generally the existence of project documentation is measured.

Context: The case study is run within an academical project at the Technical University of Vienna. During this project, a new software product is developed. The project is conducted by three doctoral students. They are planning a spin off and are going to commercialize this product. Therefore the analysis performed during the initiation phase of the extension is very important for the success of the spin off and is an excellent opportunity to test the feasibility of the extension. The following software design and software development phases are done according to the guidelines of the V-Modell XT.

- Existence and quality of the project documentation.
- Review of the results of the project and the functionality of the product. Have the initial requirement of the product been met.
- Feedback of the developers after the project is analyzed. Their attitude towards the process is captured.

The hypothesis and measures put constraints on the type of statistical test, which has to be used. Due to the fact, that the case study is concentrating on a specific project, the data is limited. Statistical analysis of e.g. the kilo lines of code (KLOC) or the fault per KLOC cannot be used, because they are not significant for the analysis of the extension of the V-Modell XT.
The use of this project as a research basis is an excellent opportunity to evaluate the extension of the V-Modell XT.

Selection of Variables

The independent variables are those variables, that can be changed during the case study. The variables should have some effect on the dependent variable and must be controllable. The independent variables and their measurement for this case study are:

- The knowledge of the team
  It is measured through questionnaires to get the skill of the single developers.

- The complexity of the project
  The complexity can only be assumed through the comparison to other projects.

- The time schedule
  The time scheduling throughout the project is measured and changes in planning are observed.

used in the development project.

The effect of the treatment is measured in the dependent variables. Often there is only one dependent variable and it should therefore be derived directly from the hypothesis. The dependent variables for this case study are:

- the feasibility and

- the efficiency

of the developed extension of the V-Modell XT.

Hypothesis measurement

During the course of the development project data is collected through weekly meetings, such as the amount of documentation, progress of the project, amount of tasks which have to be done. The variables specified in the last section are measured using questionnaire to. Also the time spent on various tasks during each phase is measured. The results of the case study are evaluated through audits to assess the success of the process and the success of the application of the created extension of the V-Modell XT in comparison to other projects, which have been performed previously. The result of the case study is presented in Chapter 7.

Hypothesis formulation

An important aspect of experiments is to know and to formally state clearly what is to be evaluated in this case study. Therefore several hypotheses for this case study are formulated according to the research questions, presented in Chapter 4.1.

1. Hypothesis: The extension to the V-Modell XT is consistent and applicable.

   The first hypothesis guarantees, that the extension to the V-Modell XT is consistent and applicable. Meaning, that possible logical errors during the creation of the extension are checked by implementing this extension to an actual project. Due to the fact, that the extension is based on already existing methods and processes the consistency and application should be possible. The expected result of this hypothesis is, that the created extension is consistent and applicable to create a new product.

2. Hypothesis: The V-Modell XT and its extension is not applicable for small project groups.

   The second hypothesis focuses on the second research question, stating, that the V-Modell XT and also the extension is an additional effort, which is quite critical in small software development companies. Therefore this hypothesis states, that the extension can not be applied to small project groups, because of the lack of resources. The expected result for this hypothesis is, that it is not possible for a small development group to successfully apply the extension to a product development project.

3. Hypothesis: It is possible to integrate an agile development process into the V-Modell XT.

   The third hypothesis checks if the integration of an agile development process in the V-Modell XT is possible. Agile development processes are especially suitable for small development teams and wherefore this hypothesis deals with the second research question. The expected result of this hypothesis is, that the integration of an agile development process into the V-Modell XT is possible. A positive answer to this hypothesis substantiates the research question, that the extension to the V-Modell XT is applicable for small companies.

4. Hypothesis: The goals of the project are all met. The project is successful.

   The last hypothesis deals with the successful completion of the project, which is performed during the case study. It is very important for the success of the extension to the V-Modell XT, that the project can be finished on time. The expected result of this hypothesis is, that using the created extension a project team is able to successfully finish their product development projects on time and within budget. Otherwise both research questions are weakened, because if the project cannot be finished on time, this extension is only another theoretical process, which cannot be realized in a real world project.

5 Product Development Process Meta-Model

The meta-model for the product development process (PDP) is based on the general product development processes, described in Chapter 2.3. The ideas for documents and products are taken from prodSEM and the concept and names for the decision-gates are taken from ABB Stage Gate Model. They are then combined to create the PDP meta-model, which also includes Business Management and Requirements from the RUP.
The three components result in a PDP meta-model. This chapter describes the meta-model and its phases and decision-gates in detail.

5.1 Overview

The meta-model basically consists of several phases, starting from the initial phase, where the idea for a new product is born, to the final phase, where 'lessons learned' from this product are reflected. Several decision gates lie between these phases. At each gate the management of the company decides, whether the development of the product should continue or not. These decisions are based on products and documents developed during the execution of each phase. If the results of the last phase are not satisfactory, it is possible, that the phase has to be repeated.
Figure 5.1 shows the basic meta-model its seven phases, its eight decision gates and the duration of each particular phase. The phases and decision gates are described in the following.

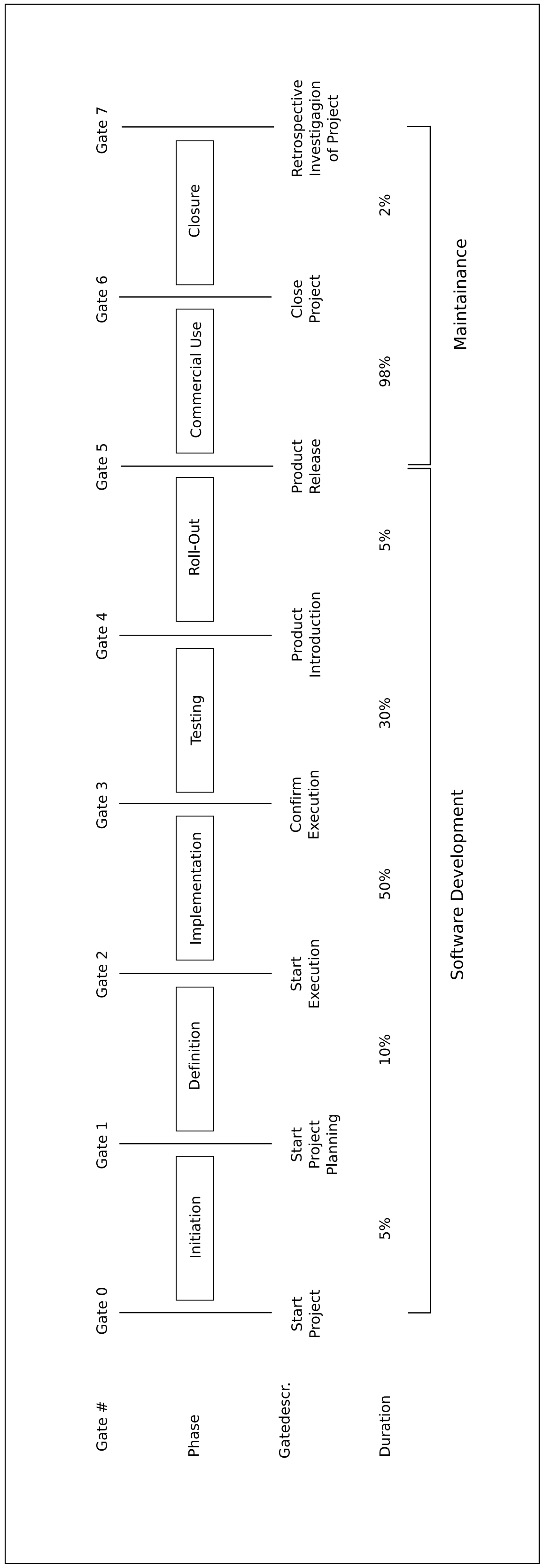

Figure 5.1: PDP Meta-Model

5.2 Phases

With reference to the research question, all phases of a product lifecycle have to be considered. Not only the beginning phases, where the main activity is done by management but also the development and closure phases have to be included. Therefore the following phases are identified.

1. Initiation
   During this phase a new idea is tested for its business value. This stage deals with the gathering information about the market, existing products, competitors, copyrights, risks, resources or what technology is to be used. The result of this stage is useful technical and business information, concerning the new product, on which the decision for further product-development can be based on. This stage should take about one percentage of the project's calculated time.

2. Definition
   This stage is the definition stage. All necessary planning is done during this phase. Requirements for the new product are defined. The project is planned, software design and resources are scheduled. The results of this stage are a complete technical specification, project- and business-plan. The product manager is responsible for this phase is and works together with a project manager, a strategic marketing manager, a technical team to develop the desired documents. During this phase it is important, to observe the market, so that the requirements and goals can be adapted appropriately. This stage should take about two percentage of the project's calculated time.

3. Implementation
   This stage covers the implementation process. Problems may arise, if a new technology is used. They should be solved during this phase. The results of this stage should be a working prototype. All risks of new technology or resource conflicts should be solved. The product should be ready for beta-testing or small adjustments. During this phase it is important, that the market still is being observed, so that the requirements and goals can be adapted. This stage should take about ten percentage of the project's calculated time.

4. Testing

   During this stage the developed prototype is tested. The result of this stage is a working product, ready for field- and acceptance-testing. This stage should take about twenty percentage of the project's calculated time.

5. Roll-Out

   This stage is the final stage before the actual roll-out to the market. The product is tested in certain areas with specially selected customers. The result of this stage is a working product according to the requirements specified in the definition phase. It has to be 'ready for the customer'. This stage should take about five percentage of the project's calculated time.

6. Commercial Use

   During this stage the product is sold and maintained. Minor adaptions, bug fixes and change requests are done. The product manager has to observe the market during this phase and react to changes, such as a new product from a competitor or changed requirements from the customers. These observations have to be documented and periodically discussed with strategic management. This stage should take about sixty percentage of the project's calculated time.

7. Closure

   During this stage all information, which has been gathered throughout the whole product development process, is processed and useful information is stored for future use in new product development processes. This stage should take about two percentage of the project's calculated time.

5.3 Decision Gates

During the course of a product creation project it can occur, that the product, in its current form is no longer profitable and before further losses are experienced the project has to be stopped. Therefore certain decision points have to be included into the meta-model. The decision gates used in this diploma-thesis are taken from [74] and are further described in the next passage.

- Gate 0: Start Project (SP) At this gate management has agreed to start the development of a new product. The result is a written explanation why the project is started, postponed or canceled.

- Gate 1: Start Project Planning (SPP) At this gate the scope of the product is agreed upon. Basic requirements are defined. The financing of the development is proposed.
- Gate 2: Start Execution (SE) At this gate the requirements and project plans are agreed upon. All requirements need to be feasible and have to be approved by the technical staff. Specification is complete, the project plan is correct and the product is ready for development.
- Gate 3: Confirm Execution (CE) At this gate the technical solution and all risks are agreed upon.
- Gate 4: Product Introduction (PI) At this gate the prototype and field-testing are agreed upon.
- Gate 5: Product Release (PR) At this gate the readiness for release is agreed upon.
- Gate 6: Close Project (CP) At this gate the termination of the product it agreed upon.
- Gate 7: Retrospective Investigation of Project (RIP) At this gate the evaluation of the project, the project's results and the product's success are completed useful information is stored to be used in new products.

This is the basic meta-model for the product planning process. Of course the meta-model can also be tailored for special needs in a company, just like other models. But the main aspects of the meta-model remains constant.

5.4 Meta-Model Summary

The following summarizes all phases, decision gates, products and responsibilities of the meta-model. The column 'Phase/Decision Gate' lists all the phases and decision gates defined in the meta-model. To the right side, the products of each phase are shown. They are subdivided into 'Input' and 'Output'. Input meaning the products or information, which will be processed during the current phase. Output lists, as the name is already suggesting, the products, which have to be available at the end of this phase. The last column, 'Responsibility', lists the roles, which are responsible for the creation respectively the processing of the input products.

Phase / Decision Gate	Input	Result	Responsibility
Gate 0 (SP)	Product Idea	Product Initiation	Strategic Management
Initiation Phase	Product Initiation	Product Proposal incl.: Market Study, Technical Study, Support Plan, Top Level Requirements, Investment Analysis, Business Plan and Business Case Analysis, Investment Analysis, Technical Specification, Project and Risk Plan	Product Manager
Gate 1 (SPP)	Product Proposal	Project Commission	Strategic Management, Product Manager
Definition Phase	Project Commission Product Proposal	Product Requirements, Project Plan, Technical Specification, Business Plan	Product Manager, Project Manager
Gate 2 (SE)	Product Requirements, Project Plan, Technical Specification, Business Plan	Project Development Decision	Product Manager
Implementation Phase	Product Requirements, Project Plan, Technical Specification, Business Plan	Product Requirements, System prototype, Risk Assessment, Test Plan	Product Manager, Project Manager, Test Manager
Gate 3 (CE)	Product Requirements, System prototype, Risk Assessment, Test Plan	Technical Solution and Project Risks are agreed upon	Product Manager, Project Manager
Testing Phase	Requirements, System prototype, Risk Assessment, Test Plan	Roll-Out Plan, Launch Plan, Working Prototype	Product Manager, Product Manager, Project Manager and Test Manager
Gate 4 (PI)	Roll-Out Plan, Product Requirements plan	Roll-Out Decision	Product Manager
Roll-Out Phase	Roll-Out Plan, Launch Plan, Working Prototype	Working Product	Project Manager
Gate 5 (PR)	Working Product, Small Change Requests	Commercial Release Document	Product Manager approved by Strategic Management
Commercial Use	Commercial Release Document, Sales Plan, Launch Plan, Marketing Plan	Change Requests, Hot Fixes, Patches, Revenue, Success Stories, Case Studies	Support Manager, Sales Manager, Marketing Manager
Gate 6 (CP)	Success Stories Revenue	Decision of Closure	Strategic Management
Closure	Decision of Closure	Closure Documents	Product Manager
Gate 7 (RIP)	Closure Documents	Product Closed	Product Manager

Table 5.1: PDP Meta-Model Overview

6 Integration of the Meta-Model into the V-Modell XT

The following section at first identifies the missing processes in the V-Modell XT, which have to be included according to the meta-model. Figure 6.1 shows the starting point of the comparison of both models. On one side there is the meta-model and on the other side is the V-Modell XT. It can be seen, that the V-Model XT already covers the whole development process, but the other parts are missing. After these processes have been identified, they are developed according to the guidelines of the V-Modell XT. This includes the role and activity descriptions and product predefinement. The result is the extension of the V-Modell XT, which covers the desired product planning process.

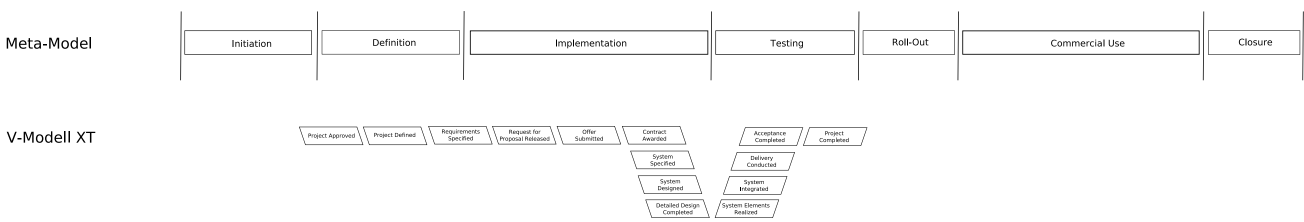

Figure 6.1: Comparison meta-model to the V-Modell XT

6.1 Meta-Model compared to V-Modell XT

This section compares the meta-model to the V-Modell XT. The processes of the V-Modell XT are compared to the process identified in the meta-model. Figure 6.2 shows the extension of the V-Modell XT compared to the meta-model. The decision gates marked in blue are the new ones inserted due to the meta-model. There are also two 'V's in this figure. The first one represents the development of a new product and the second one stands for the maintainance and evolution of this product. Both are covered by the V-Modell XT.

The project execution strategy is already defined in the V-Modell XT. All activities from the time the project is approved and the offer is accepted to the delivery of results of the project. The current V-Modell XT consequently covers the phases: Definition, Implementation and Testing. The phase Commercial Use is also covered by the V-Modell XT, if a second project is started. During this phase fixes to the product or, should the market require, significant changes to the product can be made.

6.2 The PDP Extension

The following Table contrasts the phases and decision gates of the PDP in opposite to the decision gates of the V-Modell XT.

PDP Meta Model		V-Modell XT	
Decision Gate	Result	Decision Gate	Result
Gate 0 (SP)	Product Initiation	-	-
Gate 1 (SPP)	Project Commission	Project Approved	Project Assignment
Gate 2 (SE)	Project Development Decision	Project Defined, Requirements Specified	
Gate 3 (CE)	Working Prototype, Riskplan	Detailed Design Completed	
Gate 4 (PI)	Roll-Out Decision	Project Completed	
Gate 5 (PR)	Commercial Release Decision	Contract Awarded	
Gate 6 (CP)	Decision of Closure	Project Completed	
Gate 7 (RIP)	Product Conclusion	-	

Table 6.1: Comparison of the meta-model and the V-Modell XT

Table 6.1 shows, that three phases - Initiation, Roll-Out and Closure - are not defined in the V-Modell XT and therefore have to be defined. This definition is made in the following sections. First the Roles are described, afterwards the products are defined and at last the activities are presented.

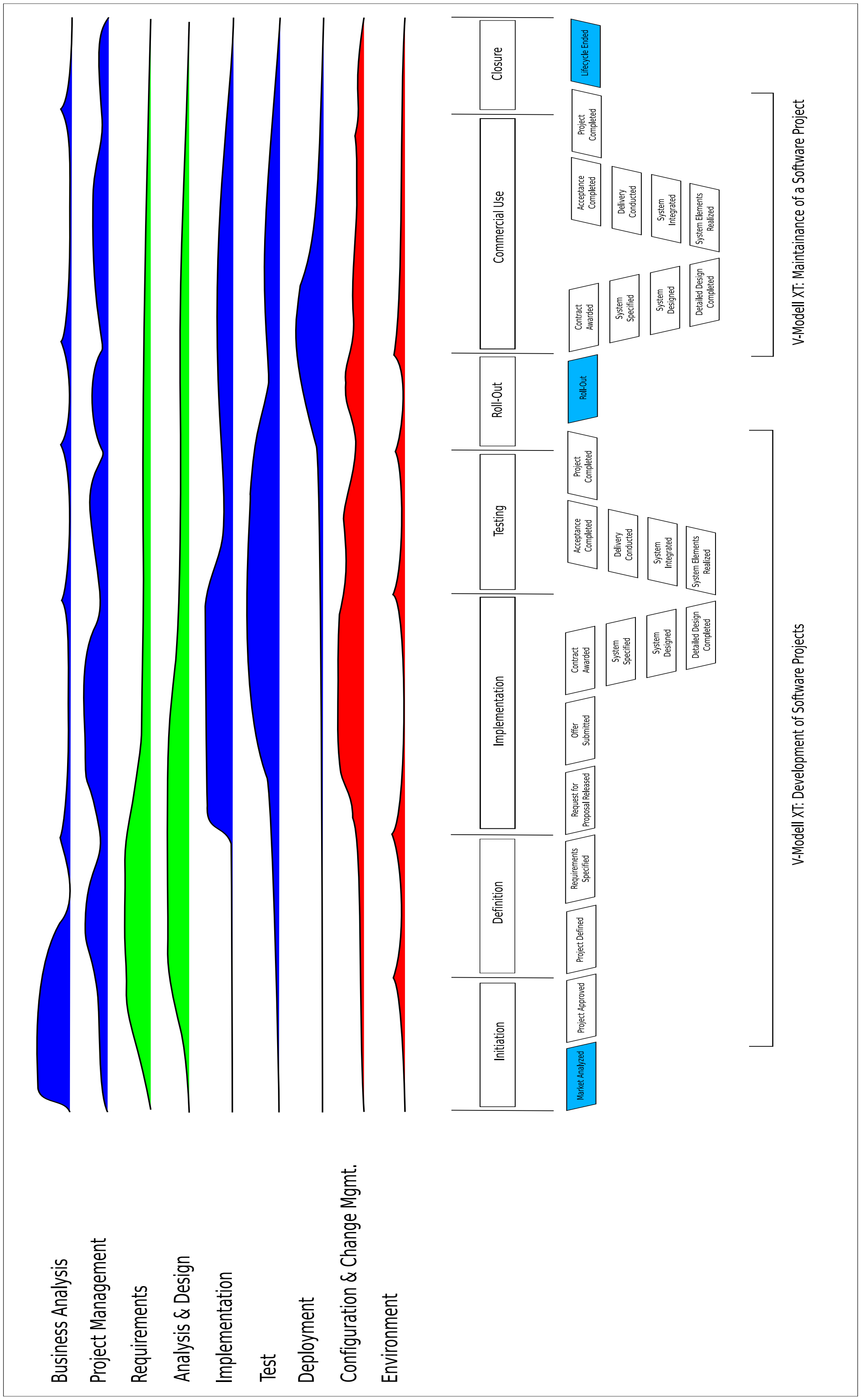

Figure 6.2: Comparison Meta-Model to V-Modell XT

6.3 Additional Roles and Products

Now that the phases and decision gates have been specified, the roles (who), activities (how) and products (what) must be defined. The following passages describe the products, activities and roles, which are essential for the successful application of the product planning meta-model.

6.3.1 Roles

This section gives an overview on additional roles which have to be introduced to cover the requirements of the meta-model. These roles are in addition to the existing roles of the V-Modell XT.

- Strategic Manager
  A strategic manager, often the chief executive officer (CEO), or a strategic management team is responsible for researching and monitoring trends in the market, observing the products and trends of competitors and developing strategies to meet all current and future competitors. Strategic management is also responsible for assessing the success of each current strategy, to quickly adapt to changed circumstances.

- Product Manager
  A product manager is basically attending products, in accordance to the strategic roadmap of the company, from the initial idea to the end of the product's lifecycle. This includes market research, product definition and design, project management, evangelization of the product, product marketing and product lifecycle management.

- Marketing Manager
  The tasks of a marketing manager are to perform market research and monitor market trends. Additionally marketing managers prepare launch events, partner events and exhibitions. They collect and communicate market information from customers or partners to internal departments and vice versa.

6.3.2 Products

This section proposes additional products, which have to be created to reflect the requirements of the meta-model.

- Product Proposal

  This document states the value of a new product and gives an outline of the possibilities of this new product. The document should also include a market study, technical study, service or support plan, high level product requirements, business case analysis, investment analysis, technical specification, project and risk plan, business plan.

- Project Progress

  This document contains all information concerning progress of the project. It is updated after each iteration, to give the product manager an overview of the features and current status of the project.

- Market Study

  This product contains an analysis of the market. One of the results should be a product/market matrix and a detailed SWOT analysis. The purpose of this product is to show, if there is volume for the new product, or to point out, that there is no chance for the new product. The market analysis document has to be recurrently updated, to quickly react to changes of the market. Status messages are communicated to the product manager

- Product Conclusion

  This document gathers all 'lessons learned' during this project. This document is created and extended during several meetings. The goal is to summarize all positive and negative experiences collected during the product's life cycle.

6.3.3 Activities

This section proposes additional activities, which have to be created to reflect the requirements of the meta-model.

- Perform market analysis

  The result of the market analysis is a document containing the current market situation, trends in the market, needs of the customers and products from competitors. To fulfill these tasks tools such as BCG Matrix and SWOT Analysis, as described in chapter 2, have to be used.

- Analyze the Product Result

  A review of all the information about the project and the product gathered during it’s lifecycle. Information collected during the meetings and project documentation is processed to create ’lessons learned’. These ’lessons learned’ are the foundations for future projects and products.

7 Evaluation of the Extension

This chapter presents the results obtained from the case study to evaluate the applicability of the extension to the V-Modell XT. A project in an academical environment has been chosen. The goal of this project is to create a new product, representing current research results. The reason why this group is using the V-Modell XT along with the newly created extension, is that it, up to now, was using agile methods to develop software products, but without proper project management and proper project documentation. With increasing complexity and maturity of their products and along with their plan to found a company, a more traceable quality standard has to be used. The project is described briefly in the next section, followed by details of its execution and the results of the case study in later sections.

7.1 Project Description

The application of scientific methods to the analysis of semiconductor devices has been challenging since the first simulation tools were developed. Due to the manifold of requirements resulting from device structures with a huge spectrum of local feature sizes as well as different physical effects this field has always been extremely difficult from a data structural as well as from a mathematical point of view. Due to the highly diverging and complex requirements in the field of technology computer aided design (TCAD) the results and applications can easily be transfered to other fields of research.
In the next decade device structures are expected to evolve substantially. Currently used manufacturing techniques must be enhanced in order to be able to produce devices delivering the desired performance improvements. Additional research concerning TCAD is therefore required. The application and development of computational methods and software tools for the design of integrated semiconductor devices and their fabrication processes is a key element to achieve the expected progress. The partitioning of the simulation domain is an essential task in order to solve the underlying equations, usually partial differential equations, with appropriate numerical discretization schemes, such as

finite differences, finite volumes, or finite elements. The selected numerical discretization scheme affects the properties of the usable spatial discretization mechanisms, such as mesh element criteria (e.g. Delaunay), topological and mesh structure (e.g. unstructured, structured topology), quantities, interpolation mechanisms, and requirements for geometry. Robustness, performance, and accuracy for these sub-parts of TCAD heavily depend on techniques and quality of this spatial discretization, the so-called mesh generation.

Automation and coupling of mesh generation and mesh adaptation for scientific computing driven by error estimation - considering the discretization technique used and the subsequent properties of the equation system - are investigated and integrated into the final product, meshX. A mesh with as small as possible number of elements, that still captures all relevant features of the investigated geometry is desired in order to support methods for fast numerical analysis as well as an accurate description of the geometry. This in particular aids the discretization of partial differential equations used for the solution using a reasonable amount of time, computer resources, and minimal manual interaction. The technical problems we have to solve in our project are mostly caused by finite numerics in the discrete scheme of computer aided design. As a consequence algorithms based on geometrical predicates have to be designed very carefully with respect to numerical issues. From a software point of view the coupling of different software modules for modeling, generation, adaptation, and error estimation merit special consideration. This requires new and more sophisticated methods of software design with special attention to robustness, orthogonality, modularity, and reusability. The application of our approach is mostly focused on semiconductor device structures, but is not restricted to this field. Final results obtained from our meshing methodology are demonstrated by real-world examples which surpass the capabilities of currently used meshing techniques.

Project Details

The project in located in the field of software development with an academic background. It is scheduled for 21 man-month. The project start was on the 05.01.2007 and it ended on the 05.08.2007. The costs of the project are about 70.800 EUR and are explained in Table 7.1. The project team consists of three persons, which already have experience in this field from similar projects respectively from their dissertations in the field of Scientific Computing, Simulation Methodologies and Higher Order Discretization Schemes. The list of the project's milestones can be seen in Table 7.1. The main objective of the project is to develop the product meshX with the following features:

Project start	05. January 2007
Project end	05. August 2007
Project team	3 Persons
Project costs	Total costs: EUR 70.800

Table 7.1: Basic Project Information

Project start	05.01.2007
Design case studies are finished	07.03.2007
Development, experiments and tests are finished	03.07.2007
Project end	05.08.2007

Table 7.2: Project Milestones

- an input/output module for different formats (CSG, STL, HIN, VOL)
- a surface mesh generation module
- a volume mesh generation module
- a module for the resampling steps
- an interface for other meshing algorithms

Non-Objectives of the project were:

- a working installer for the product

- solve portability issues for different operating systems (Windows, MacOS, Linux)
- the creation of a webpage and a commerce system for the sale

The non-objectives will be covered in the post-project phase respectively in other projects. Figure 7.1 shows an example of the output of meshX. On the left side is the input mesh and on the right side is the mesh, created by meshX. It can be observed, that the meshX structure is more straightforward, than the input structure.

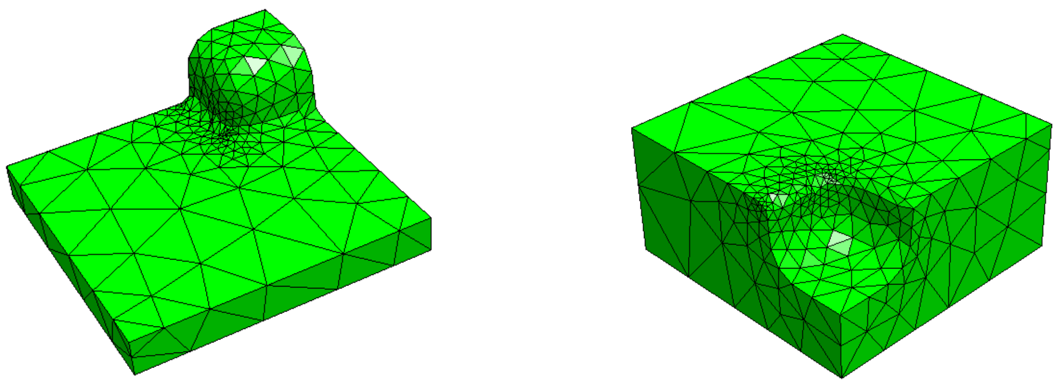

Figure 7.1: Meshing output of meshX. Two examples from device simulation.

7.2 Operation

In the following the operation of the case study is presented. This section is divided into three parts - the preparation, the execution and the presentation of the results.

7.2.1 Preparation

All participants of the case study are working at the Technical University and have done projects prior to this one. They are highly qualified and committed to their work and were informed of the case study.
All materials, which are important to the case study, are created during the course of the project. The analysis and verification of the hypotheses is based on these materials.

7.2.2 Execution

The case study was executed over eight months during which three persons, as already stated in the project introduction, worked on this project. Data was primarily collected through forms. Short feedback was also given by each of the team members during weekly meetings. Interviews were used at the end of the case study, to evaluate the course of the project. The feedback included the current progress of the project, the created products during the last iteration and their impression of the extension. Feedback for improvement of the model was appreciated. The progress was assessed on a regular basis and differences from the plan were immediately communicated and integrated into project panning. The resulting products are evaluated and discussed regarding their completeness. The results of the case study are presented in the following and are discussed in Chapter 8.

7.3 Results of the Case Study

The case study is motivated by the still present problem, that software products are delayed or even canceled. This situation is know for year and despite the introduction of several well known quality standard software products still get postponed or even canceled. One major problem is the missing communication of Management and Engineering within one company. This diploma thesis investigates two research questions. This first research question deals with the cooperation of management and engineering within one company - 'Is it possible to combine Management and Engineering into one process¿ By including management and engineering in one process during the lifecycle of a product, the quality and the possibility that the product is finished on time and within budget increases. Most of the time management, which performs market and competitor analysis, comes up with the idea of a new product. They have a vision, what the product should look like and which features should be included. Engineering on the other hand does not have this big picture concerning the product and therefore has no idea on which feature they should concentrate on respectively which part of the product they should implement most efficiently. Management and engineering have to work together and the final goal must be well known throughout the company, to give everybody the opportunity to contribute to the result to their best of their abilities. This can be accomplished by increasing the cooperation of management and engineering. This diploma thesis not only investigates this very issue but it also suggests a possible approach to this issue.

The second research question focuses on the application of the new extension to the V-Modell XT and of course the application of the V-Modell XT to the processes of small companies. Most often these companies only consist of a few people and, while already performing their work as good as possible, these companies often have to struggle to meet deadlines of projects and to fund the resources for further projects. It is very common for small companies arise from a great new idea for a new product but with no funding. The founders of these companies invest all of their money in the foundation of the company and the development of this very product. Small software companies often lack proper resource planning and therefore spend their money suboptimally. They may have some kind of project management, but this project management is often limited to todo lists and some initial time planning. Resource planning therefore is crucial to small software companies, due to their limited resources. The V-Modell XT and the created extension require some effort to be implemented properly. This effort is very critical in small software companies, because of the described situation. The benefit of the implementation of the V-Modell XT and its extension is the proper management of new respectively existing products and a guideline for successfully managing the creation of new products, on which companies inexperienced with product management also can rely on.

These main research questions as starting point, the following four hypotheses are identified.

7.3.1 Hypothesis 1 - Feasibility

The first hypothesis guarantees, that the extension to the V-Modell XT is consistent and applicable. Possible logical errors during the creation of the extension are checked by applying this extension to an actual project. Due to the fact, that the extension is based on already existing methods and processes the consistency and application should be possible. The expected result of this hypothesis is, that the created extension is consistent and applicable to create a new product.

During the course of the project it turned out, that the requirements were not as specific as needed. Therefore the specification and the requirements were refined and all tasks were adapted. This adaptation of the development could easily be made because of the use of an agile development process and this benefit is further explained during the results of hypothesis three.

Another problem was the well know fact, that customers are realizing what they really want during the course of the project. This also happened during this project. The customer needed some additional features, which were a bonus to the new product. Therefore the product has more features than initially planned. Management recognized this opportunity and incorporated the additional features. Also this adaptation could be incorporated because of the agile development process.

Another benefit of using the extension to the V-Modell XT, which is one of the most important reasons for the small software development company performing this case study, is the creation of project and software documentation. Table 7.3 give an overview of documents created during the course of the project. It can be seen, that up to this project, the software development company, was only partly creating necessary documentation respectively necessary project and product planning. This information is based on interviews with the project members of the development team. Most of the products were not needed in former projects, so they were simply not created. The team did research on some topics, but they never did write it down respectively create a document to assure the quality and traceability for later analysis. This changed during this project. The development team did create several important documents, such as the assessment of strengths

and weaknesses or the classification of their future products in a product portfolio matrix. To assure that all changes and decision remain traceable in the future, all decisions were traced using the project management tool in combination with a project plan.
Through interview, performed with the project manager and the team members of the development team, it turned out, that the team and their development strategies lack of priorization. Their applied software processes were most of the time ad hoc, meaning that the often had no proper time respectively resource planning. As described in [83], some of the symptoms of inadequate portfolio management apply to the company under inspection.

Phase	Product	Other Projects	This Project
Business Analysis	Strength and Weakness comparison (SWOT Analysis)	-	+
Business Analysis	Portfolio analysis (BCG Matrix)	-	+
Business Analysis	Revenue estimation	-	+
Product Planning	Software Design, Requirements Document	-	+
Project Planning	Project plan	-/+	+
Definition	Requirement Specification, Software Design Specification	-/+	+
Development	Sourcecode Documentation	-/+	+
Testing	Automated tests	-/+	+
Documentation	Software Documentation	-/+	+
Documentation	Project Documentation	-/+	+
Support	User Manual	-	+

Table 7.3: Product comparison between other projects and this project. (+ .. created, -/+ .. partly created, - .. not created)

7.3.2 Hypothesis 2 - Small Project Teams

The second hypothesis focuses on the second research question, stating, that the V-Modell XT and also the extension is an additional effort, which is quite critical in small software development companies. Therefore this hypothesis states, that the extension can not be applied to small project groups, because of the lack of resources. The expected result for this hypothesis is, that it is not possible for a small development group to successfully apply the extension to a product development project.

Figure 7.2 and Figure 7.3 depict the time calculated and the actual time spent on the different tasks respectively different phases of the product creation process. This shows, that the creation process still was more important to the development team, than the analysis and definition phase of the development process.

Phase	Time planned	Time spent
Initiation	150h	85h
Definition	450h	405h
Implementation	1700h	2045h
Testing	1700h	1575h
Roll-Out	200h	70h
Commercial Use	0h	0h
Closure	0h	0h
Sum	4200h	4180h

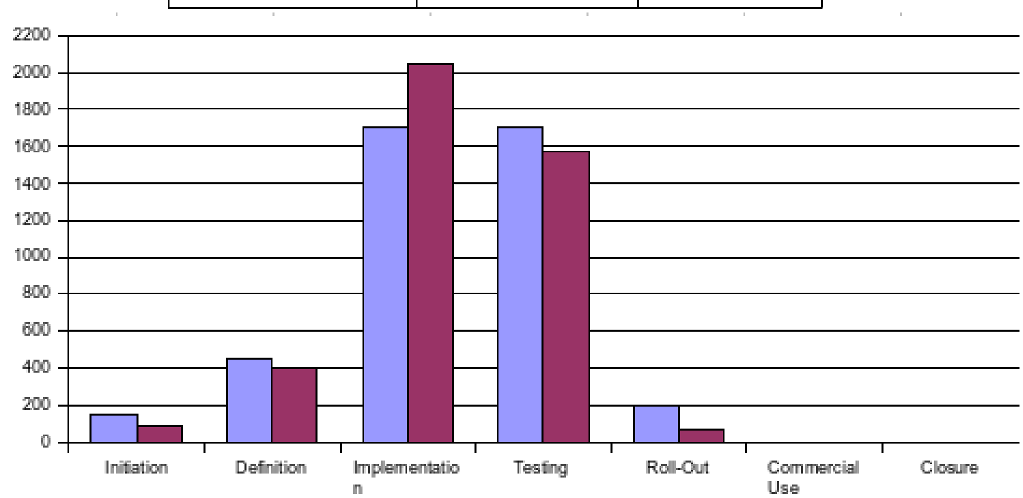

Figure 7.2: Overview of time spent on different phases

7.3.3 Hypothesis 3 - Agile Development Process

The third hypothesis checks if the integration of an agile development process in the V-Modell XT is possible. Agile development processes are especially suitable for small development teams and wherefore this hypothesis deals with the second research question. The expected result of this hypothesis is, that the integration of an agile development process into the V-Modell XT is possible. A positive answer to this hypothesis substantiates the research question, that the extension to the V-Modell XT is applicable for small companies.

Task	Time planned	Time spent
Business Analysis	100h	85h
Product Planning	100h	85h
Project Planning	200h	180h
Definition	150h	140h
Development	1300h	1570h
Testing	1300h	1100h
Documentation	900h	950h
Support	150h	70h
Sum	4200h	4180h

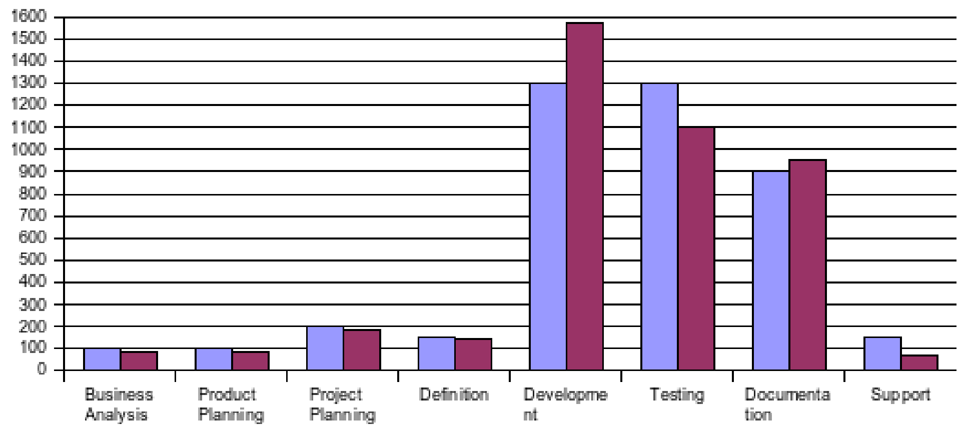

Figure 7.3: Overview of time spent on different tasks

The application of Extreme Programming as agile development method was realized successfully. The development team was able to use Extreme Programming as their main development process. Using an agile approach the team was able to react on changes which occurred during the corse of the project, such as changing requirements and additional features by the customer.

Unfortunately it is very common for software development projects that requirements are to fuzzy at the beginning of the project and become more and more tangible during the development process. The problems mentioned in the introduction, that the requirements were missing and fuzzy, and that during the course of the project the customer did come up with new requirements, could be solved quite easily. Because of the weekly meetings where planning for the next iterations is done, the team could flexibly react on changes. This resulted in a slight increase in the time spent on development, but the overall time constraints were met. These weekly meetings are a benefit of the application of an agile development strategy.

7.3.4 Hypothesis 4 - Successful Project

The last hypothesis deals with the successful completion of the project, which is performed during the case study. It is very important for the success of the extension to the V-Modell XT, that the project can be finished on time. The expected result of this hypothesis is, that using the created extension a project team is able to successfully finish their product development projects on time and within budget. Otherwise both research questions are weakened, because if the project cannot be finished on time, this extension is only another theoretical process, which cannot be realized in a real world project.

But in general the project was finished on time and provided the required product using the created extension to the V-Modell XT. The product included the expected features and produces the expected results.

The project team also encountered several problems. These problems were the same problems as described in the introduction and experienced in nearly every software project: the problem of a missing or fuzzy requirements specification. These problems have already been mentioned during the presentation of the results of the hypotheses.

Another important change was the use of a project management tool. An open source software tool named dotProject [77] satisfied the requirements of the development team. Using this software tool, the development team was able to track all relevant project data. This was one of the major improvements the team experienced during this project. The traceability of tasks, the estimation of effort and revenue also was important during project planning.

The main problem identified is the quite tight time planning. Because of the team size all additional work is difficult and time consuming. Statements of the team: This project was more work, but we believe this additional effort really shows off in the future.

8 Discussion

In this chapter the research questions and the hypotheses stated in Chapter 4 are assessed and compared to the results of this diploma thesis and the results of the performed case study. First the results of the case study are discussed. Afterwards the findings from the case study are compared to the underlying research questions of this diploma thesis.

The hypotheses derived from the research questions are revised in the following:

1. Hypothesis: The extension to the V-Modell XT is consistent and applicable.
2. Hypothesis: The V-Modell XT and its extension is only applicable for larger project groups.
3. Hypothesis: It is possible to integrate an agile development processes into the V-Modell XT.
4. Hypothesis: The goals of the project are all met. The project is successful.

The main focus of this diploma thesis is pointed to the questions whether it is possible to narrow the gap between management and engineering, by using a single development model throughout the product creation process. The second main question is whether it is it possible to apply this model to the development processes of a small company.
As a global outcome the case study shows, that the extension is working and that even a small team can make use of a traceable product management process unsing the extension. All hypotheses are proven except the second one. The result is in contrast to this hypothesis, but this result is even better, because the extension to the V-Modell is hereby not limited to larger development teams. The following will further analyze the results of the case study.

8.1 Hypotheses Analysis and Interpretation

Four hypothesis have been investigated, which are shown in Table 8.1. The project ended without delay and within the defined time and budget. The following compares the hy-

Hypothesis	Result
The extension to the V-Modell XT is consistent and applicable.	Hypothesis is affirmed. The development group was able to use the extension.
V-Modell XT and the extension is only applicable for larger project-groups.	No, also applicable for a small project-team.
It is possible to integrate an agile development process into the V-Modell XT.	Hypothesis is affirmed. The integration is feasible.
The goals of the project are all met. The project is successful.	The hypothesis is affirmed. All requirements of the project were met.

Table 8.1: Comparison between the initial hypothesis and the results of the case study

pothesis and the result of the evaluation of the data from the case study. It can be observed, that all the results are positive, meaning that the developed extension of the V-Modell XT is actually working and that it is also possible to apply this extension to a small development team respectively a small company. But it also has to be mentioned, that the circumstances of the project may not be applicable to other projects. The context and especially the people involved in this project were an important success factor. This result shows, that the project team using the extension has to be highly motivated respectively it is an enormous advantage if the project team is highly motivated.

Hypothesis 1 - Feasibility

The first hypothesis is confirmed. It is possible to apply the created extension of the V-Modell XT to an product development project. This can be proved by the fact that the project is finished successfully, meaning that the planned features and functionality was developed correctly. Also the existence of project documentation, as seen in Table 7.3, is an important factor of the success of the extension. Another benefit of this extension to the V-Modell XT is the explicit performance of the business analysis of the new product. This feature is especially interesting for small companies, because they often have no product management in place. Using this extension, these companies are guided through the product creation process, from the initial idea to the final product. The main focus, bringing Management and Engineering together in using one process model, can be done using this extension to the V-Modell XT. They have to communicate respectively agree on the products of the

Hypothesis 2 - Small Project Teams

The second hypothesis has to be rejected. The development team was able to apply the extension to the product development project. Through interviews with the project manager and the project team, it turned out, that during this project they spent much more time on documentation than they used to spend on former projects. The time planned and the time spent on the single tasks and the phases can be seen in Figure 7.2 and Figure 7.3.
The management of the company is convinced, that their project and product management needed major improvement. Using the extension additional to the V-Modell XT give management a tool to manage they project and product portfolio of the company. The company under investigation is not only able to drastically improve the quality of the development project, but now they have classified their projects, products and product ideas using portfolio management. This gives them an opportunity to efficiently plan development resources in the future.

Hypothesis 3 - Agile Development Process

The third hypothesis, the integration of an agile development method into the V-Modell XT, is confirmed. The development team was able to use Extreme Programming, as suggested by the extension, in combination with the V-Modell XT. Using Extreme Programming it was able to react on changes flexibly. The problems mentioned in Chapter 7, that the requirements were missing and fuzzy, and that during the course of the project the customer did come up with new requirements, could be solved quite easily. Because of the weekly meetings where planning for the next iterations is done, the team could flexibly react on changes. This resulted in a slight increase in the time spent on development, but the overall time constraints were met.
It did turn out, that the extension is capable of dynamically reacting to changes. This is because of the flexibility of Extreme Programming and the constant evaluation of progress and tasks which have to be done for the next iteration. Changed requirements and added features can be handled using this approach.

Hypothesis 4 - Successful Project

The fourth hypothesis is also confirmed by the results of the case study. Following the guidelines of the extension in combination with the V-Modell XT, the development team was able to finish the project on time and within budget. As already mentioned, even short-term changes in the requirements are handled by the the extension to the V-Modell XT. Their lack of product and project management was remedied using this model.

8.2 Research Question Analysis and Interpretation

The last section discussed the four hypotheses of this case study, whereas this section discusses the research questions. It can be summarized, that the outcome of the case study substantiates the two research questions from Chapter 4.

The first research question - ‘Is it possible to create one process model, which can be used by management and by engineering, to successfully create products?’ - can be answered positively. The results of the first and forth hypothesis proofs this statement, but with the restriction, that the development team has to be highly motivated. It is possible to integrate management processes and models into the V-Modell XT. With the help of this extension, shown in Chapters 5 and 6, this combination is possible.

The second research question - ‘Can this extension be used by small teams respectively small companies to develop their products?’ - can also be answered positively. This statement is substantiated by the results of hypotheses two and three. The case study shows, that even a small team can perform and finish a product development project on time and within budget according to the extension of the V-Modell XT.

9 Conclusion and Outlook

Originating from the problem, that software products nowadays are still postponed or even canceled, or that customers do not get what they wanted. One problem is the communication between management and engineering. They have different fields of expertise and are therefore using different vocabulary. This misunderstanding leads to wrongly understood requirements, late product releases or even product cancellation. Another common problem is the missing portfolio management in small companies. Often they are convinced, that they are too small to apply portfolio management and have no resources for this task. The solution proposed in this diploma thesis, combines models from both worlds into one already existing model. Management and engineering processes are combined into the V-Modell XT. At first the requirements for a general product development process have been defined and put into a meta-model. This meta-model is then compared to the V-Modell XT. The missing parts are identified and are worked out. The result of this steps is an extension to the V-Modell XT. Using this extension, the V-Modell XT covers the whole product creation process.

The result of this diploma-thesis, the extension of the V-Modell XT, is then evaluated using a product creation project at the Technical University of Vienna. The goal of this project is to create a product, which includes the knowledge of the research from the past three years. Due to the fact, that the product will be released, the development team needed a more mature quality assurance and therefore used the V-Modell XT in association with the new extension.

The result of the project is as expected. A small team or a small company is able to successfully finish a project using the V-Modell XT and the newly developed extension. There is overhead for creating the necessary documentation, but it is this overhead which ensures the quality and documents all decisions during the development of a project. Another benefit, which closes the gap of missing portfolio management in small companies, is that the extension already has such tasks included in its processes. Therefore small companies using this extension, are at the same time applying portfolio management to

their projects.
The main statement of the evaluation is, that the extension is working and that small companies respectively small development teams are able to integrate the V-Modell XT and the extension into their development process. Using this extension, product managers are able to keep track over the products in development. Management and engineering are induced in working together much closer by using the created extension, which leads to successful finished product development projects.

It is an enormous benefit for a company to establish only a single process model, which covers the whole process from the initial idea of a new product and also includes the development, the commercial use and finally to the decision to end the product's lifecycle. The V-Modell XT not only offers this but also makes all of this traceable by management. Project managers are able to track the progress of the development process using the V-Modell XT, while the same model is used by the engineers, who implement and develop the product. Product requirements, defined by management, can be translated into technical requirements for the developers. Quality managers oversee the overall quality of the product creation process. Therefore the resulting product is identical to the required one, due to the guidelines provided by the V-Modell XT and its extension.

A possible outlook is to create an application, which covers all the aspects proposed in this diploma-thesis. The formal integration of the proposed extension into the V-Modell XT would also be possible.

List of Tables

3.1 The Extreme Programming way [2] 55
3.2 Summary of the presented models 58

5.1 PDP Meta-Model Overview 72

6.1 Comparison of the meta-model and the V-Modell XT 74

7.1 Basic Project Information 81
7.2 Project Milestones 81
7.3 Product comparison between other projects and this project. (+ .. created, -/+ .. partly created, - .. not created) 86

8.1 Comparison between the initial hypothesis and the results of the case study 91

List of Figures

1.1 What the customer wanted. [71] 5
1.2 Research Goal: Up to now management and engineering use their models only in their field of expertise with only a little input from other departments. This creating a gap between both disciplines. 7

2.1 Boston Consulting Group Growth-Share Matrix [18] 18
2.2 SWOT Analysis 21
2.3 Strength - Weaknesses comparison ([55] page 247) 22
2.4 Division of the product life cycle into four sections [5] 23
2.5 Overview of ABB Stage Gate Model [74] 26
2.6 Overview of Siemens prodSEM [86] 28

3.1 V-Modell XT Structure [1] 31
3.2 Classification of Projects and Subdivision into Project Types [1] 31
3.3 Process Modules and their Components [1] 32
3.4 Project Execution Strategies to Project Types [1] 33
3.5 Project Execution Strategy, Decision Gates and Products [1] 33
3.6 Decision Gates [1] 34
3.7 UP: The four phases and their milestones [28] 35
3.8 Unified Process [15] 36
3.9 Rational Unified Process [16] 39
3.10 Scrum Phases: Figure taken from [84] 49
3.11 Agile Unified Process [21] 51
3.12 XP Processmodel [19] 54

4.1 The basic idea of the research goal. 61

5.1 PDP Meta-Model 68

6.1 Comparison meta-model to the V-Modell XT 73

6.2 Comparison Meta-Model to V-Modell XT 75

7.1 Meshing output of meshX. Two examples from device simulation. 82
7.2 Overview of time spent on different phases 87
7.3 Overview of time spent on different tasks 88

Bibliography

[1] V-Modell XT english Version Teil 1

[2] Extreme Programming from a CMM Perspective, Mark C. Paulk, Software Engineering Institute, IEEE 2001

[3] www.extremeprogramming.org, visited January 2007

[4] Strenghtening the Case for Pair Programming, L. Williams et al., IEEE Software, vol. 17, no. 4, July/Aug. 2000, pp. 19-25

[5] Produktmanagement, Dieter Lennertz, F.A.Z-Institut für Management-, Martk- und Medieninformationen GmbH, 2006

[6] A Process Approach for Senior Management Involvement in Software Product Development, MŁlardalen University, 2003

[7] Strengthening the Case for Pair Programming, L. Williams, RR. Kessler, W. Cunningham, R. Jeffries, Journal IEEE Software 2000

[8] All I really need to know about pair programming I learned in kindergarten, L. Williams, RR. Kessler, Journal ACM 2000

[9] Agile software development methods: Review and Analysis. Espoo, Finland: Technical Research Centre of Finland, VTT Publications 478 Available online http://www.inf.vtt.fffpdf/publications/2002/P478.pdf, Abraharnsson P., O. Salo, J. Ronkainen, and J. Warsta,

[10] New Directions on Agile Methods: A Comparative Analysis, Pekka Abrahamsson, Juhani Warsta, Mikko T. Siponen and Jussi Ronkainen. IEEE 2003

[11] Agile Modeling: Effective Practices for Extreme Programming and the Unified Process, Ambler S., New York: John Wiley & Sons, Inc. New York,2002.

[12] A Manager's Introduction to Rational Unified Process(RUP), Ambler S., 2005

[13] V-Modell XT german Version 1.2, 2005

[14] Product management, Internetlink: http://en.wikipedia.org/wiki/Product_management, visited January 2007

[15] Unified Process, Internetlink: http://en.wikipedia.org/wiki/Unified_Process, visited January 2007

[16] Rational Unified Process, Internetlink: http://en.wikipedia.org/wiki/Rational_Unified_Process, visited January 2007

[17] Projektmanagement, Patzak G., Rattay G., Linde Verlag Wien, 2004

[18] The Boston Consulting Group Matrix, Internetlink: http://www.cipher-sys.com/HofHelp/Bcg/bcghelpfile.htm, visited January 2007

[19] Agile Software-Entwicklung, Carsten Dogs, Timo Klimmer, MITP-Verlag/Bonn, 2005

[20] Advanced Development Methods. SCRUM Development Process, Schwaber Ken,

[21] The Agile Unified Process (AUP), Internetlink: http://www.ambysoft.com/unifiedprocess/agileUP.html, visited September 2007

[22] Extreme programming and agile processes in software engineering, Baumeister H., 6th international conference, XP 2005, Sheffield, UK, Springer Verlag, 2005

[23] Extreme Programming Explained: Embrace Change, Kent Beck, Addison-Wesley 1999

[24] Planning Extreme Programming, Beck K., Fowler M., Addison-Wesely 2000

[25] Optional Scope Contracts, Kent Beck, Dave Cleal, unpublished, 1999

[26] Teaching Team work: An extreme work for first-year programmers in Extreme Programming and Agile Processes in Software Engineering, Becker-Pechau P., Breitling H., Lippert M., Schmolitzky A., Proceedings of the 4th International Conference, XP Springer Verlag pp. 386-393, 2003

[27] A Spiral Model of Software Development and Enhancement, Boehm B., Barry W., IEEE Computer, May, pp. 61-72, 1998

[28] The Rational Unified Process: An Introduction 3rd Edition, Kruchten Philippe, Addison Wesely, 2004

[29] Industrial Software Metrics Top 10 List, Boehm B., IEEE Software 4(5): September 1987, 84-85

[30] Get Ready For The Agile Methods - With Care, Boehm B., Computer, vol. 35, pp. 64-69, 2002

[31] CMM Implementation Guide, Caputo K., Addison-Wesely Longman, 1998

[32] Software Engineering Institute: Appraisal Requirements for CMMI, Carnegie Mellon University, Version 1.2, August 2006

[33] Agile Software Development, Cockburn A., Addison-Wesley, Boston 2002

[34] Winning at new Products, Cooper Robert G., Perseus Publishing 2001

[35] Large-Scale Agile Software Development, Crocker Ron., In press 2003

[36] The Agile Manifesto, Fowler M., Highsmith J., Software Development, 28-32, 2001 http://agilemanifesto.org

[37] Principles of Software Engineering Management, T. Gilb, Wokingham, UK: Addison-Wesley 1993

[39] Agile Software Development: The Business of Innovation, Highsmith J., Cockburn A., Computer, vol. 34, pp. 120-122, 2001

[40] Agile Software Development with Scrum, Schwaber Ken, Beedle Mike, Prentice Hall 2002

[41] The Unified Software Development Process, Jacobson I., Booch G., Rumbaugh J., Addison-Wesley, Reading, MA, 1999

[42] Extreme Programming Installed, Jeffries R., Anderson A., Hendrickson C., Addison-Wesely, 2001

[43] Extreme programming and the software design course in Extreme Programming Perspectives, Johnson D. H., Caristi J., In Proceedings of XP Universe, 2001

[44] CMMI, Kneuper R., DPunkt 2006

[46] Software entwickeln mit extreme Programming. Erfahrungen aus der Praxis., Lippert M., Roock S., Wolf H., Dpunkt.verlag 2002

[47] Agile Software Development: Principles, Patterns and Practicies., Martin R. C., Pearson Education 2002

[48] Rapid Development, McConnel S., Microsoft Press, 1996

[49] Comparison of CMM level 2 and eXtreme programming, Nawrocki J. R., Walter B., Wojciechowski A., 7th EuropeanConference on Software Quality, Helsinki, Finland, 2002

[51] Software-Produkt-Management, Hans-Bernd Kittlaus, Christoph Rau, Jürgen Schulz, Springer Verlag, 2004

[52] Produktmanagement, Professor Werner Pepels, R. Oldenbourg Verlag München, 2006

[53] Software-Produktmanagement, Harry M. Sneed, Martin Hasitschka, Maria-Therese Teichmann, dpunkt Verlag, 2005

[54] Product Lifecycle Management, John Stark, Spring Verlag London, 2005

[55] Erfolgreiches Produktmanagement, Klaus J. Aumayr, Gabler Verlag, Wiesbaden 2006

[56] The Unified Process Inception Phase, Scott W. Ambler, Larry L. Constantine, CMP Books, Lawrence USA 2000

[57] Designing Software for Ease of Extension and Contraction, David L. Parnas, IEEE Transactions on Software Engineering, Vol. SE-5, 2, March 1979

[59] Extreme Progamming from a CMM Perspective, Paulk M. C., IEEE Software 18(6): November/December 2001

[60] Progressive Acquisition and the RUP: Comparing and Combining Iterative Processes for Acquisition Development, Pitette G., The Rational Edge, Novemeber 2002

[61] Lean Software Development - An Agile Toolkit, Poppendieck M., Poppendieck T., Addison Wesely 2003

[62] Using XP for Safety-Critical Software, Poppendieck Mary with Ron Morsicato, Cutter IT Journal 15(9): September 2002

[63] The Scrum software development process for small teams, Rising L., Janoff N. S., IEEE Software, vol. 17, pp. 26-32, 2000

[64] Von ISO 9001 zu TQM, Thaller G. E., VDE-Verlag, 2001

[65] Software-Qualität, Thaller G. E., VDE-Verlag, 2000

[66] A Simple Model of Agile Software Processes, Vanderburg G., ACM Press, 2005

[67] Software Qualitätsmanagement in der Praxis, Wallmüller Ernst, Carl Hanser Verlag, 2001

[68] Quality Software Management (all volumes 1-4), Weinberg G., Dorset House, 1992-1997

[69] COCOMO: Estimating Software Project Effort and Cost, Caine A., Piducks A. B., IEEE 2004

[70] Handbuch Produktmanagement: Strategieentwicklung - Produktplanung - Organisation - Kontrolle, Sönke A., Gabler Verlag 2002

[71] http://www.johnsimonds.com/wp-content/uploads/2006/10/what the customer wanted.jpg, visited May 2007

[72] http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/89920, visited June 2007

[73] Experimentation in Software Engineering - An Introduction, Claes Wohlin, Per Runseon, Martin Höst, Magnus C. Ohlsson, Anders Wesslén, 2000 Kluwer Academic Publishers

[74] A Process Approach for Senior Management Involvment in Software Product Development, Christina Wallin, Mlardalen University, 2003

[75] Software Engineering Economics, Boehm B., Prentice Hall, Englewood Cliffs, NJ, 1981

[76] Study of Softwareproject results, http://www.infora.de/set.php?subject=projektnews_17, 2007

[77] Open Source Project Management Tool - dotProject, https://www.dotproject.net, 2007

[78] Procter & Gamble Co. website, http://www.pg.com.eg/history2.cfm, 2007

[79] Harvard Business School website, http://hbswk.hbs.edu/archive/1476.html, 2007

[80] Portfolio Management for New Products: Picking the Winners, Robert G. Cooper, Scott J. Edgett, Product Development Institute, Ancaster/Ontario CA, 2001

[81] Best practices for managing R&D portfolios, Robert G. Cooper, Edgett S.J. & Kleinschmidt E.J., Research-Technology Management, 41, 4, July-Aug. 1998

[82] New product portfolio management: practices and performance, R. G. Cooper, S. J. Edgett, E. J. Kleinschmidt, Journal of Product Innovation Management, July, 1999

[83] Do small software companies need portfolio management?, Jarno Vähäniitty, Software Business and Engineering Institute, Helsinki University of Technology, 2006

[84] Scrum Overview, http://www.mountaingoatsoftware.com, 2007

[85] Performance Aspects of a DSEL for Scientific Computing with C++, R. Heinzl, P. Schwaha, M. Spevak and T. Grasser, Institute for Microelectronics, Technical University Vienna, 2006

[86] prodSEM - Siemens Product Planning Process, Copy from the Technical University of Vienna