Professional MBA
Entrepreneurship & Innovation

Uncertainty unveiled – Decisions and operations in agile product lifecycle management

A Master's Thesis submitted for the degree of
"Master of Business Administration"

Supervised by
Univ. Prof. Dr. Christopher Lettl

DI Martin Suntinger
0405478

Vienna, July 1, 2011

Professional MBA
Entrepreneurship & Innovation

Affidavit

I, **Martin Suntinger**, hereby declare

1. that I am the sole author of the present Master's Thesis "Uncertainty unveiled – Decisions and operations in agile product lifecycle management", 97 pages, bound, and that I have not used any source or tool other than those referenced or any other illicit aid or tool, and
2. that I have not prior to this date submitted this Master's Thesis as an examination paper in any form in Austria or abroad.

Vienna, July 1, 2011

____
DI Martin Suntinger

Abstract

In a fast-moving marketplace characterized by high uncertainty and frequent change, finding the balance between harvesting returns from core products and innovating in parallel is a tightrope act. A tense field of competitive forces and interests drives decision making on product roadmap priorities.

We researched product lifecycle management (PLM) at the example of a software firm operating in a highly dynamic, global environment. Empirical evidence confirmed that making transparent and mutually accepted roadmap decisions, optimizing resource utilization throughout the complete product lifecycle, dealing with uncertainty and emergent information and balancing flexibility with pursuing long-term plans are key success factors. Despite the existence of numerous approaches towards product and innovation management, product strategy and operations, we identified the need for an integrative framework to aggregate relevant decision parameters and align decisions with underlying operations.

We propose a PLM framework built around a ranking mechanism of projects on the roadmap as the focal point of decision making and operations. Based on different decision perspectives such as financial valuation, market needs, technologies or stakeholders' interests, the ranking is established and continuously updated. Both a fast, semi-structured approach and a formal decision analysis method using the analytical hierarchy process (AHP) are proposed. The ranking is the basis to dynamically schedule activities based on resources and constraints. Principles of lean production are used to manage global workflows. Innovation controlling metrics are applied to monitor the execution and refine the decision process.

The evaluation showed that ranking is a powerful tool to enforce, communicate and execute clearly defined priorities. The framework seamlessly aligns decisions and operations and combines plan-driven operations with agility for dealing with exogenous and endogenous uncertainty.

Table of Contents

Index of Figures

Abstract. *The balance between harvesting returns from core products and innovating in parallel is a tightrope act. Decisions over the product roadmap and the allocation of a firm's resources are complicated by a high number of uncertainties and a tense field of competitive forces driving the decision process. Efficient decision making linked seamlessly with operations is key to succeed in a fast-changing marketplace.*

1 INTRODUCTION

1.1 Overview and challenges

Today's fast-changing and ever evolving market environments require companies to constantly innovate, adapt quickly to new industry trends and respond to competitive pressure while still harvesting returns from their core products. James March identified this tightrope act as a balance between exploration and exploitation [59], which manifests in practice, amongst others, in strategic decision making over projects and resources in product roadmap planning and steering. The product roadmap is the focal point for decision making around which products will be developed and enhanced, and how many resources will be assigned for development. Practitioners in this field find themselves confronted with a large number of stakeholders attempting to impact product directions and strategic resource allocation, and an even larger number of decision parameters and factors to consider. These factors include financial constraints, market signals, customer behavior and requirements or technology advances. Finding the balance between resource investments for maintenance and enhancements of core product lines and new products, evaluating and deciding over upcoming opportunities, or investing wisely in research for potential radical innovations are only few of the manifold challenges. The decision process is characterized by the high level of uncertainty associated with all relevant decision parameters. Required resource levels have to be forecasted and the accuracy of effort estimates typically evolves through the course of a project, with more information being gathered. At the time of making decisions, empirical evidence suggests that a significant share of estimates particularly for novel products deviate strongly from the final resource investments required. At the same time, expected returns and market estimates, against which the effort is evaluated, are highly uncertain. Often, these are based on samples and generalizations as well as assumptions. Furthermore, they are strongly dependent upon which metrics are applied for measuring success (financial returns on the specific new product, financial results overall, impact on corporate brand perception etc.). Adding to that, additional project risks and potential external or internal threats may reveal over time and heavily impact a project's success such as unforeseen operational difficulties, technological advances of competitors, new research findings or emergent technologies substituting an existing solution. These risks can be discounted in financial planning, but their quantification is subject to uncertainty.

The complexity is not only inherent to product roadmap decision making, but as Crawford and Di Benedetto stated, "*the complexity of operations and decisions is the most dramatic hallmark of product innovation*" [25]. In fact, it appears that exactly this interface - the tight link of decision making and operational execution - is one of the key success factors to master the complexity of product innovation in an environment characterized by large uncertainties. While offering vast potential for a firm to differentiate and outperform competitors, we observed that missing or inefficient structures and processes in this area may have several negative consequences, such as long cycle times, lack of quality, insufficient product features, operational inefficiencies and rework as well as frustration and constant pressure rather than a sustainable innovation pace. Operations need to be seamlessly aligned with strategic decision making, in order to populate any decision instantly to current operations and in turn feed any relevant change and information back to decision makers in order to readjust if necessary.

It is relevant to notice that in practice, decision making in new product development (NPD) activities is complicated by not only managing one project from idea to launch, but a set of parallel projects, each potentially being in a different stage of its product lifecycle and requiring different types of decisions and resource investments. Ideally, an R&D department manages to keep a continuous in-stream of ideas and project proposals, and in parallel manages and develops existing products. In case of a product roadmap, this in-stream will be a mix of different classes of ideas. For instance, requests from existing customers are a relevant source of input for product enhancements and responding to these requests has positive impact on customer satisfaction and product maturity. Emergent technologies might be adopted by enhancing existing products or creating new products and providing complementary offers. Furthermore, projects with different character might be proposed, such as projects with direct customer benefit and revenue potential serving customer needs better than existing products or fulfilling newly emerging needs, projects with indirect benefits (e.g. on brand, positioning, appeal to investors, IP portfolio etc.), projects increasing usability, demo-ability, performance or other relevant metrics or projects in direct response to competitive pressure. For choosing certain projects over others, a tense field of competitive forces has to be considered for determining priorities. The overall portfolio of projects shall be optimized in terms of catering to a maximum number of requestors and balancing short-term delivery with pursuing long-term strategic goals. Figure 1 illustrates this field of driving forces impacting product roadmap decision, opposed by the need for a consistent product strategy, aligned with corporate objectives.

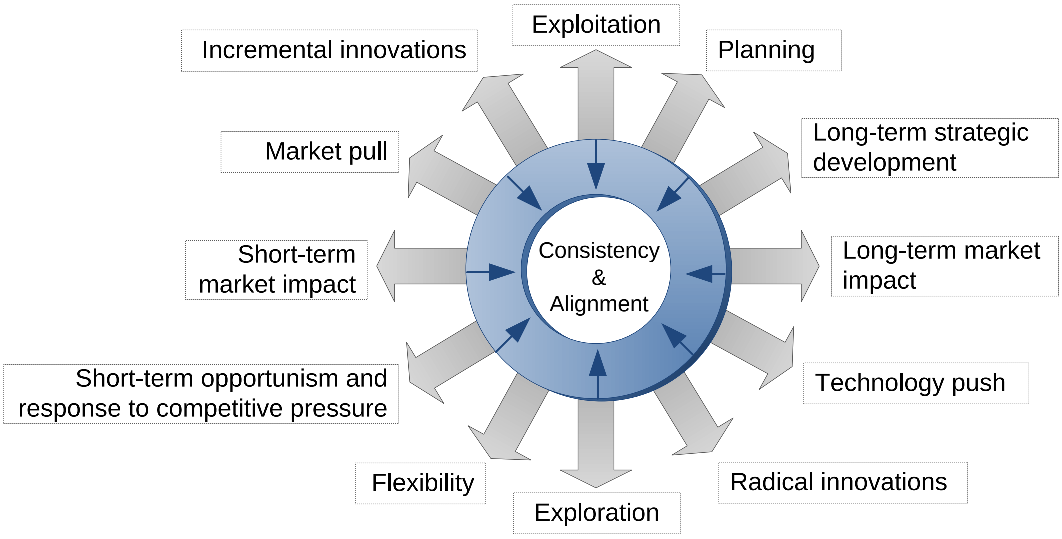


Figure 1 - Driving forces in product roadmap decisions

In research and practice, these challenges have been addressed from different perspective. Methods have been proposed for optimizing the decision making over R&D projects with mathematical or stochastic models capable of determining the ideal allocation of resources based on currently known information, involved uncertainties and resource constraints. Other approaches attempt to reduce the level of uncertainty associated with relevant decision parameters, for instance by improving forecasts' and estimates' accuracy and methods to collect early market feedback. Others argue that operations are the key to master these challenges and proposed methods to streamline NPD processes and lower the involved risks through incremental resource commitments and strict decision processes, or to establish flexible product development processes allowing making late changes in order to react to emergent requirements. Additionally, approaches such as rapid prototyping, test marketing, user-oriented design, or advanced forecasting techniques are used to reduce market uncertainty. On the technical planning side, various methods have been proposed towards effort estimations in order to reduce uncertainty and deviations in the resource planning. Undoubtedly, these methods are able to increase the accuracy of input parameters to the roadmap decision process problem and provide valuable input for prioritization and project valuation, but how can all these decision factors be aggregated to a consistent roadmap? How are decisions populated to and aligned with established processes in operations? We believe it requires an efficient and easy-to-use decision model and decision processes, aligned with operations at all stages of the product lifecycle to ideally leverage a firm's resources and capabilities for bringing products to the market.

1.2 Objectives and research question

This thesis focuses on the question of how to make product roadmap decisions and align these decisions seamlessly with operations in an environment characterized by high exogenous and endogenous uncertainty and frequent change. In this context, relevant related aspects are researched, including which factors drive the decision process, how these potentially conflicting decision factors can be aggregated, how to set up the operational framework to execute on frequent decisions and dynamic prioritization, and how to measure and control the overall process.

The objective of this thesis then is to develop a practical framework for product lifecycle management under uncertainty. The framework shall provide concrete guidance on how to structure the product roadmap decision and planning process, which decision parameters to consider, how to aggregate all decision parameters, and how to setup and schedule operations, which tools, roles and responsibilities are needed to incrementally gather relevant information and execute on roadmap decisions.

1.3 Research field and method

This thesis studies the field of strategic product planning at the example of an Austrian software company. The firm currently employs around 250 people, of which approximately 40% are in R&D. Due to its highly dynamic character, the software industry has taken a lead role in many recent developments in the area of new product development. Thus we consider it a relevant and attractive sample to conduct our research. The research for this thesis has been carried out in several stages:

Stage 1 – Case study and analysis
Based on quantitative data (effort estimates and resource investments over the product lifecycle) and qualitative interviews with key stakeholders, the current product planning and development processes are analyzed in order to identify decision drivers, characteristics of the organizational setup, potential weaknesses and bottlenecks. These empirical findings are complemented by an extensive literature review on related research findings.

Stage 2 – Model development
Based on the analysis results and existing literature, a novel framework for product roadmap decision making and continuous planning is elaborated.

Stage 3 – Model Implementation and evaluation
The framework is evaluated based on observations from the practical implementation of key aspects and problem-centric interviews with relevant stakeholders in the company.

1.4 Research Contribution

This thesis contributes to the fields of NPD and strategic product management a new framework, which links the strategic level of optimized decisions over the product portfolio to the operational level, providing practical guidance on how the decision making framework can be applied in practice, which roles, responsibilities and processes are required, as well as how these can be synchronized with existing development processes. The model is developed based on quantitative data and qualitative interviews in a context of an SME in the software industry.

Abstract. *In literature, the complexity in product lifecycle management has been tackled from different perspectives and touches a broad field of specialized research areas, including R&D project and portfolio management approaches, NPD and innovation management processes, and strategic management of NPD. All of these areas provide valuable contributions and building blocks, which we adopt and integrate into a holistic view on product lifecycle decisions and operations.*

2 LITERATURE REVIEW

The problem of optimizing technology roadmap decisions has manifold dimensions Thus, the literature on this topic is very broad and touches several specific research branches, depending on which perspective is taken (see Table 1). The discussion below attempts to summarize the most relevant concepts and findings in each of these fields, together with their benefits and shortcomings addressed below in the proposed approach.

Focus on decision making and planning	Focus on operations

▪ Project portfolio optimization
▪ Product and project management
    ▪ Agile product management
▪ Strategic management of NPD and innovation
    ▪ Methods for customer-oriented design*
▪ Financing and controlling of innovation
    ▪ Real-options analysis and thinking
▪ Multi-criteria decision analysis*
	▪ NPD and innovation management
	    ▪ Specific to software engineering
	        ▪ Lean and agile software development
	    ▪ General NPD processes and techniques
	        ▪ Flexible product development
	        ▪ The fuzzy front end of NPD

**Relevant concepts from these research areas, which we integrate into the approach but which do not contribute to the overall model structure will be discussed below in the course of the proposed approach.*

Table 1 - Overview of related research fields

2.1 Project portfolio optimization approaches

From a high-level perspective, the roadmap decision process is a selection problem, i.e., identifying the ideal set of projects and enhancements from a pool of candidates in order to maximize the output based on a defined metric such as return on investment (ROI). R&D project portfolio optimization approaches offer the theoretical foundation for optimizing the choices among potential R&D projects. Different models have been proposed to model the uncertainties inherent to this decision problem, such as real-options analysis [6][7][31][57], linear/stochastic programming [86] or simulation-based approaches [5]. It has been shown that from a theoretical perspective the portfolio optimization problem is NP-hard [86]. In practice, however, the applicability of such models is constrained by efforts for data input and modeling to apply such frameworks, and the inherent uncertainty associated with all said decision parameters. Thus, results of these methods are only as good as the processes and people involved in incrementally gathering, modeling and quantifying the available information on expected or remaining development efforts and future returns. Despite these limitations, relevant findings can be drawn from research in this area. In particular, Bardhan et al. [6] argue that for IT projects the understanding of project dependencies combined with real-options analysis leads to significantly different valuation and portfolio results in cases where certain projects can be leveraged for the implementation of others (e.g. via reusable platforms/modules).
Due to said limitations, in this thesis R&D portfolio optimization approaches are not applied. Instead, a pragmatic, incremental decision process is proposed. It takes specific aspects such as options valuation into consideration where necessary, but does not rely solely on finding a closed, algorithmic solution.

2.2 Product and project management

Both the product and project management disciplines have been studied extensively in literature and contribute valuable concepts to roadmap decision making. We selected Haines' reference book for orientation in this field [38]. Haines categorizes product management activities into three areas: new product planning, new product introduction and post-launch product management and names "making decisions" as one of the core activities of a product manager. According to Haines the traditional product management literature relies on a combination of forecasting, scenario analysis, and business cases based on discounted cash-flow analysis to decide about and justify projects. Various decision making techniques are proposed, including *combining options*, the *morphological box*, *decision matrices* and *decision trees*. Whereas the traditional product management literature focuses on plan-driven development approaches, recent work in the field discusses product management in the context of flexible and agile development processes. Goodpasture puts agile, iterative NPD processes originating from the software domain into an enterprise context and proposes different planning cycles for a project from time-boxed iterations to releases, to waves, to projects [37]. This view on different planning cycles is adopted and applied to the proposed framework and extended by a more detailed discussion on how to actually make decisions and manage operations within these cycles. Additionally, the author proposes an agile business case built on different granularity levels. While valuable for communication, the actual project valuation does not leverage the advantages of agile and adaptive development methods and the gained flexibility. Out of these considerations, our approach incorporates ideas of real-options analysis to account for

these characteristics.
In parallel to seeing an adoption of agile practices in the product management discipline, also the agile software development world starts incorporating product management perspectives into agile practices. Pichler [68] discusses the role of product management in association with agile SW development, particularly with the Scrum [80] process. Several valuable concepts are presented, such as product visioning techniques (cf. Reid and De Brentani who studied the effects of market visioning on product performance [71]), focusing on the minimal marketable product, or a product backlog which is essentially a list of features detailed on demand by priority. What remains unconsidered is how this backlog is managed effectively in practice, how prioritizations are done, how changes happen, and which information is required and taken into consideration. Similarly, Cohen [21] discusses agile techniques for product managers. Among other techniques, he outlines the use of value stream mapping [58] to identify "waste" and measure process cycle efficiency in the development process. This technique is taken as a reference and extended for innovation controlling in the on-hand model.
In summary, the product and project management disciplines offer countless tools for managing and optimizing plan-driven and predictive NPD. When it comes to agile and adaptive methodologies, methods and techniques are still emerging and do not provide a holistic view on the enterprise. Only specific aspects such as the development process are covered in depth. The validity of these methods has been proven in successful real-world projects, but they leave sufficient leeway to better leverage the characteristics and benefits of agile thinking when being applied on enterprise level.

2.3 NPD and innovation management

Reducing cycle time and thereby speeding time to market has been discussed as one of the most important management goals in product development. Crawford and Di Benedetto summarize several techniques for gaining speed and differentiate between time to market and time to success, which includes also post-shipping speed [25]. Parry et al. [65] studied the impact of NPD strategy, product strategy and NPD processes on perceived cycle time and found success factors in all three areas, including a formal NPD strategy or creating a climate for innovation, whereas heavyweight project management had no significant impact on perceived cycle time.

2.3.1 Prioritization and decision making

In the innovation management literature, decision making and prioritization under uncertainty have been discussed by several authors. A valuable summary on existing approaches has been presented by Goffin et al. [34] who tie the aspect of decision making closely to the principles of portfolio management and summarize concepts for project valuation and portfolio balancing. First, the authors distinguish between one-step projects, multi-step projects and network projects allowing numerous decisions along the project lifecycle. Depending on the project type, different valuation techniques are applied, including basic DCF (discounted cash flow) analysis coupled with a sensitivity analysis, probability-based decision trees, Monte-Carlo simulation, scoring systems, and real-options analysis. The authors favor a multi-criteria scoring system and argue that real-options analysis did not prove valuable for a majority of studied innovation projects. Furthermore, the authors propose establishing a project ranking based on these valuations. The aspect that different projects will

require different (or the same) resources in the implementation is left up to management to be resolved and optimized. In a second step, Goffin et al. propose to balance the project portfolio based on an analysis of strategic fit, time and resources and a risk/reward matrix.
A similar approach is proposed in this thesis. Yet, in contrast to Goffin et al., we claim that optimizing resource utilizations over the complete product portfolio is possible without mathematical black-box solutions, and we propose a structured decision process to establish a more sophisticated project ranking, not only based on the financial valuation but also the technology, market and stakeholders' perspective.
Tidd and Bessant [89] propose managing uncertainty through an incremental commitment of resources. By generating knowledge, uncertainty can be converted into calculable risk, which can then be addressed. Furthermore, the authors propose building a broad portfolio of projects to spread risks, and summarize several tools assisting with decision making under uncertainty. These include analyzing alternative futures based on forecasting, scenario analysis and trend extrapolation, prototyping, probing and learning, i.e., to make small exploratory steps into uncertain areas, applying alternative measurement and evaluation criteria, mobilizing support networks, using alternative decision-making tools, deploying alternative funding-and implementation structures and mobilizing entrepreneurship. Yet, the authors provide no concrete guidance on how to implement these generic proposals in practice in different industries. Many of these techniques are implicitly incorporated in the proposed PLM model and extended by a concrete guidance for the practical implementation (see section 4).

2.3.2 NPD processes

Saren [77] identified seven distinct categories for NPD processes (see appendix 7.1 for a short summary of all models). Out of these, activity-stage models and recent network models are most relevant for this work. Activity-stage models, as their name suggests, focus on individual activities rather than functions, and allow for iteration and feedback loop. Popular examples for activity stage models are the Booz, Allen and Hamilton model of new product development [13], and one of the most wide-spread NPD process models nowadays: the Stage-Gate© model devised by Cooper [22]. It divides the NPD processes into six subsequent stages from discovery to product launch with intermediary decision gates allowing discarding or continuing a project. Such models are able to reduce project risks by an incremental lock-in of resources into a project and systematic decision making processes. Yet, once certain stages such as the design are run through, it is hard to adjust to emergent requirements or early customer feedback, meaning there is flexibility for decisions only at the predefined gates, but not throughout the complete process. In a more recent update published by Cooper [24], some of the limitations are addressed and refined. Amongst others, Cooper argues that Stage-Gate© is neither necessarily a linear process, nor a rigid system, while at the same time proposing better governance methods and "gates with teeth". Less bureaucracy and leaner decision gates, clearly defined gatekeepers and methods for scaling the process to adjust to different types and sizes of projects have been proposed. Further updates include flexible and adaptable versions of Stage-Gate© achieved via spiral development and simultaneous execution, better decision-making practices including scorecards, success criteria, self-managed gates, electronic and virtual gates, and integration with portfolio management. Finally, a more rigorous post-launch review and integration with open innovation methods have been introduced (see also Grönlund and Rönnberg [36]). Despite these updates, the major pitfall of the method is that certain opportunities might be wiped out at the decision gates too early. Future options which might result from a particular project are not considered and explored.

In contrast, network models are built on the idea of constant knowledge accumulation from various inputs, either from internal or external sources. A network of external linkages is coupled seamlessly with internal development. An illustration of the model can be found in the appendix, section 7.1. The model developed in the course of this thesis builds on the concept of a network model, which we tailor to the particular needs of the given context.

2.3.3 Flexible product development

Aside of these basic NPD models, methods of flexible product development have been proposed, and particularly widely adopted in the software industry under the umbrella term of agile software development. Early work in this area was published by Wheelwright and Clark [99], Smith and Reinertsen [81][82], Iansiti [46] and MacCormack et al. [60]. Flexible product development aims at reducing time to market based on overlapping development phases rather than a sequential, linear process. Decisions are pushed to the latest responsible moment in order to keep flexibility as long as possible. Smith defines flexibility in product development as "*the ability to make changes in the product being developed or how it is developed, even relatively late in the development, without being too disruptive.*" [59]. Thompke and Reinertsen [88] introduced a flexibility index, which expresses the ratio of a change in a perturbing variable to the change in projected lifecycle costs. The lower the costs of change for a particular product attribute, the higher the flexibility. Agile software development processes achieve this by many short-lived iterations of design, development and testing, finishing a modular part of the final product in each iteration. With the ability to constantly make changes, these models acknowledge the uncertainties associated with technical planning and market forecasting and allow adjusting to emergent information and network changes. Several concepts play a key role in flexible product development, including a focus on modular design (enable changing individual parts without effecting others, isolate parts subject to higher uncertainty or growths), set-based design (focus on constrains rather than a path of development), real-options thinking, time-boxing to steer priorities and a focus on decisions and risks rather than tasks and processes.

Still, most of these approaches are focused on the operational level rather than the strategic level and give no guidance on the decision making frameworks required to leverage the agility [12]. These methods offer the operational tools to quickly adjust to changed priorities, emergent requirements and early market feedback, but the decision processes and frameworks to gather the required information and make continuous prioritization and resource allocation decisions are often undefined black-box models, happening implicitly in the mind of a talented product manager.

2.3.4 NPD in the software domain

Since the publication of the agile manifesto in 2001 [9], two major paradigms for NPD are dominating the field of software development: Traditional, plan-driven approaches, which are predictive in nature, versus agile, adaptive methods. Examples for plan-driven approaches include Cleanroom

[63], Capability Maturity Model (CMM) [66] and Team Software Process (TSP) [96]. On the other end of the spectrum, popular agile methods are Scrum [80], Adaptive Software Development (ASD) [41], eXtreme Programming (XP) [8], Crystal [20] or Feature-Driven Development (FDD) [64]. Boehm and Turner [12] discuss these competing paradigms and provide a comparison based on the scope and characteristics of the methods. In brief, plan-driven approaches focus on process improvement and capability, risk management and verification/validation techniques as well as typically extensive, detailed planning in advance, comprehensive specification and documentation. In contrast, agile methods are designed to embrace change, have frequent delivery cycles, rely on tacit knowledge, and favor individuals and interactions over processes and tools [9]. The authors propose to use risk assessments as the basis for selecting one method over the other, and also identify several common grounds. MacCormack and Verganti's study [61] on selecting a suitable process for a given context yielded a similar result: The authors propose to refer to technology and market uncertainty for matching process and context. If uncertainty with the platform being developed and the targeted market is low, traditional activity-stage models are likely to be successful. In contrast, if uncertainty is high, flexibility is required and can be achieved by *reacting* to new information. MacCormack and Verganti contribute mainly on a conceptual level by arguing that the process must always match a given and varying context, and that there is no single best-practice approach. Boehm and Turner focus on the development processes and with that on the operations perspective on software engineering. In this work, this balance between adaption and prediction is analyzed from a more strategic perspective and generalized for product roadmap decision making and product lifecycle management. We claim that throughout the product lifecycle, the context and thus the process needs change and need to be managed differently.

More recently, a growing body of literature emerges alongside of agile methodologies, discussing the application of lean production in software development. Furugaki et al. [32] discuss the implementation of the Toyota Production System (TPS) for software engineering and claim that methods of traditional manufacturing can be applied and mapped to this field. Ladas [50] discusses the use of Kanban pull-based systems for lean software development and describes a process model which incorporates Kanban into the agile Scrum software development process. Core ideas of value-orientation, batch size management and a pull mechanism for scheduling operations are applied similarly in this work to both the operations and the decision management in roadmap planning. Further relevant discussions on lean software development concepts can be found in Poppendieck and Poppendieck [69] who describe key lean principles such as elimination of waste, late decisions, amplification of learning, and the concept of local responsibilities vs. global optimization, as well as in Shalloway et al. [84] or Larman and Vodde [51] who discuss scalability of lean and agile development for large projects.

One of the major white stains in NPD in the software domain is the post-launch phase. Processes are designed to manage the development until the software is released. Yet, empirical research by April and Abran [4] suggests that software maintenance accounts for 50%-90% of product lifecycle costs. The authors summarize software maintenance best practices and an adjusted maturity model which discusses the specifics of software maintenance. We claim that the currently dominant approach of separating post-launch maintenance from NPD has several downsides: It increases the effort for in-depth knowledge transfer between different teams, maintenance becomes discontinuous, meaning that there is only rare feedback into the NPD process, and maintenance resources are bound and cannot be leveraged on-demand for NPD tasks if the fluctuating maintenance effort is low at a point in time. Furthermore, it is unfeasible for smaller teams, where exactly agile methods emerged and claim to have most significant benefits. In contrast to this view, all post-launch activities, including maintenance and support tasks are considered as part of the product lifecycle in the resource planning and scheduling.

2.3.5 The fuzzy front-end of NPD

One particular aspect of NPD is the so-called fuzzy front end of innovation, a term first coined by Reinertsen in 1991 [70]. It refers to the earliest phases of an innovation process, before being structured at an organizational level. According to different authors, activities in this stage include problem structuring and identification (Leifer et. al [52]), information collection (March [59]) or early stages of concept development (Cooper [23]). A more detailed study of the fuzzy front end theory is out of scope for this thesis. Yet, referring to Reid and De Brentani [70] who quote several studies in the field, evidence exists for a correlation of activities in the fuzzy front-end phase with later product success. Therefore, this aspect shall be considered in the proposed model without contributing novel findings to this research area. In the proposed approach, the theoretical model of Reid and De Brentani [70] is used as a foundation. It identifies three perspectives: the environment needed to foster activities at the fuzzy front-end, the individuals and their roles, and the organization. Two core roles have been identified, i.e., inventors and ruminators. Inventors do have a technology vision and ruminators bring the market vision in the early-stage activities to promote and structure an idea. Their environment allows individuals to link corporate and individual knowledge with information from their environment, which may result in innovation.

In section 4 we will discuss how these aspects can be integrated into the product lifecycle management framework.

2.4 Conclusions

In this section, numerous related concepts and approaches have been discussed to address the challenges in roadmap decisions and operations. Most relevant streams of research are the agile and flexible product development domain which relies on establishing maximal process flexibility to *react* as effectively and fast as possible, and the traditional product and innovation management doctrines, which rely on *predictive* analysis techniques, sophisticated project valuation methods and a balanced portfolio of projects. These basic streams are supplemented by different NPD models, and research on the roots of innovation, with the fuzzy front-end model currently being the most relevant theoretical foundation.

In the next section, results from an empirical case study at a software firm are discussed. As often, advances in research are not necessarily fully implemented and reflected in practice. We attempt to capture both views: the theoretical foundations and challenges in the practical implementation of proposed models.

Abstract. *Qualitative research of product roadmap management practices and a quantitative analysis of R&D investments over the product lifecycle at a software company confirmed the inherent challenges to this process even in a successful firm. Empirical evidence supports the assumption that making transparent and mutually accepted roadmap decisions, optimizing resource utilization throughout the complete product lifecycle, dealing with uncertainty and emergent information, balancing flexibility with pursuing long-term plans, and incentivizing all stakeholders towards optimizing the overall R&D output are key aspects to be tackled in an effective product lifecycle management approach.*

3 DATA ANALYSIS AND OBSERVATIONS

In the course of this work, R&D processes and decision making have been studied at the example of UC4 software, a vendor of enterprise software for IT automation headquartered in Austria. The results will serve as the empirical basis and evidence for the problem fields addressed by the proposed PLM framework. The analysis consists of two parts:

1) Qualitative analysis of roadmap decision drivers and NPD processes – Based on observations and problem-centric interviews with key stakeholders the current product lifecycle management framework is analyzed in order to understand the current decision process and decision criteria applied, and identify perceived weaknesses.
2) Quantitative analysis of R&D resource investments – Based on historic records on initial estimates and actual development and maintenance efforts the quality of the existing roadmap decision framework is assessed in order to identify potential deviations and shortcomings. In particular, the aspect of initial costs of development versus lifecycle costs and their role in the planning process is illuminated.

In this section, key findings and conclusions from the case study are summarized. Due to corporate policies, only condensed research results could be published from the quantitative data analysis. Detailed data on individual projects and R&D investments need to remain undisclosed. Yet, for the purpose of this thesis the high-level summary of findings is sufficient to understand key issues in the product lifecycle management process.

3.1 Qualitative research results

Qualitative research was targeted towards understanding the product roadmap decision process (Who is currently involved in product roadmap decisions? How are decisions carried out? What is the planning and decision horizon? How flexible is the roadmap planning process? How is it integrated with NPD operations?), and the NPD operations (Which development processes are applied? How is the interaction between roadmap decisions and operations being implemented?).

3.1.1 The product roadmap: stakeholders, roles, and processes

At the time this research was carried out, the roadmap planning process in the studied firm involved the following key stakeholders, and their duties were defined as follows: The product marketing department is ultimately responsible for the roadmap and its presentation for approval by the board of directors. Product marketing is furthermore responsible for analyzing the market needs and business case for all potential projects. This function is close to the more frequently encountered product management function. The product design department is in charge of providing a high-level concept of each project as a basis for the later functional design. Product development is responsible for judging technical feasibility and providing effort estimates for the business case. The board of directors approves the product roadmap for a period of 18-24 months and is involved in regular status update meetings on the progress of the implementation as well as for approving changes to the original plan. After the initial approval of the product roadmap, product design and program management (which in this case is defined in one role) coordinates the development stages from concept creation to quality assurance. Product marketing coordinates the product launch. Technical post-launch maintenance is managed within the product development and support group without the involvement of product management. Product design is involved only in maintenance cases which require functional product enhancements. The applied NPD model can be categorized into the group of activity-stage models, adapted to the specifics of software development.

The long-term product roadmap for a planning period of 18-24 months defines one major release cycle of the software. Intermediate, minor releases are scheduled within the frame of the major release. The scope for the major release is fixed at the beginning of the planning period. It is proposed based on input from product design, product development and product management, whereby product management proposes projects and focus topics from a market perspective, product design creates rough functional concepts and product development provides preliminary effort estimates used for the planning process. In addition, product design and development propose roadmap topics based on technical advances, platform features, ideas within the team, or enhancement requests from customers. Once the scope of the roadmap is fixed, major changes (e.g., abandoning one project in favor of another) need to be approved in a regular roadmap steering committee meeting which takes place every two weeks. Participants include product management, development VPs the CTO and regularly also a representative from the board of directors.

In the regular process, after roadmap changes have been approved in the roadmap steering committee, these can be implemented starting from the next development iteration. Thus, an average timeframe from making a roadmap change that requires approval to the beginning of its implementation is between 2 and 4 weeks (up 2 weeks until the next roadmap committee meeting,

and then up to 2 weeks until the next planning iteration in development takes place). If the change requires additional design work, it might take even longer until it finds its way into product development.
Beyond the timeframe of 18-24 month, only a rough product agenda exists in product marketing, which defines high-level topics to be addressed within a period of 3-5 years, without any further details or concrete actions associated with these, besides the product roadmap described above.

3.1.2 Roadmap decision criteria

From problem-centric interviews with these stakeholders, the following criteria and techniques emerged as the main decision drivers (priority in the given order):

1) Direct revenue potential (ideal situation: pre-commitments of customers available)
2) Competitor analysis: Comparison of strengths and weaknesses
3) List of informal criteria (e.g. strategic fit, competitive pressure, visibility of impact, etc.)

We observed a rather informal decision process. Projects with customer commitments and visible, direct revenue potential are generally preferred due to the lower level of market uncertainty as compared to other projects. Then, projects filling gaps in comparisons against competitors or strategic gaps in the product positioning are preferred. Finally, a list of informal criteria is applied to argue for other issues, whereas these are typically qualitative presentations and the decision process may well vary depending on how strong a project is "promoted" by its supporters. Overall, the decision process relies less on hard facts and numbers (aside of the direct revenue potential vis-à-vis the development costs) but rather on an informal list of identified benefits that can be realized with the project from a customer perspective and a strategic perspective. This picture was completed by a quote of the CEO, who commented on decision analysis tools as *"[…] tools exist to make better gut-feeling decisions*".

3.1.3 NPD processes

At UC4 Software, at the time of writing this thesis different product development processes are applied in different development groups. The largest group developing the core product line applies an adapted multi-level V-model [30] as illustrated in Figure 2. This model is particularly wide-spread in Central Europe and its variations (V-Model 97, V-Model XT) are used as a standard process in publicly funded, governmental projects in Germany and other countries.

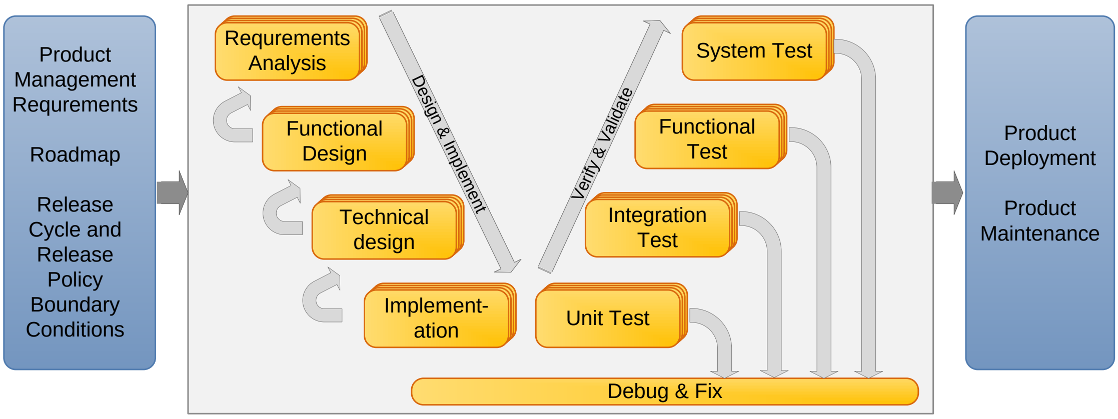


Figure 2 - Multi-level V-model at UC4 Software

Source: UC4 Software, adapted from internal process documentation

As can be seen from the illustration, the model focuses on the design, implementation, and testing phases. The product roadmap and development priorities are expected as process inputs from product management. Post-launch product deployment and maintenance are also not covered in the original V-model and are performed by separate teams.

In other development groups, the agile Scrum [80] development process is applied. Key concepts are so-called development sprints, the product backlog and the product owner and Scrum master roles. Sprints are iterations of 2-4 weeks starting with a planning meeting where tasks are assigned among the team members and ending with a sprint retrospective. Communication is fostered by a daily stand-up meeting where every team member shares the status, relevant information or impediments with the team. The product backlog is a list of features to be developed. From the backlog, the most important items are moved to the sprint backlog for the next sprint. The product owner maintains the product backlog, the Scrum master ensures the process, communication and facilities required.

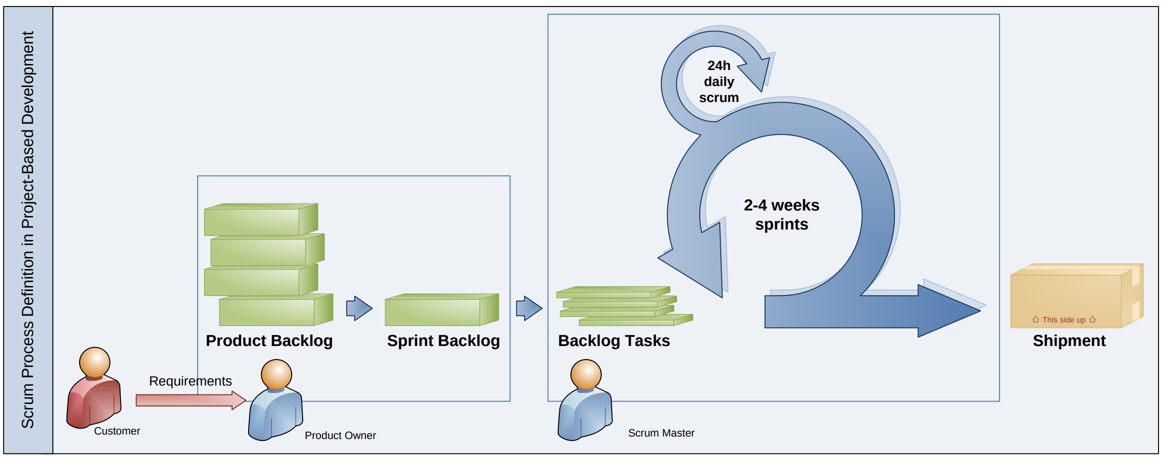


Figure 3 - Original definition of Scrum process model

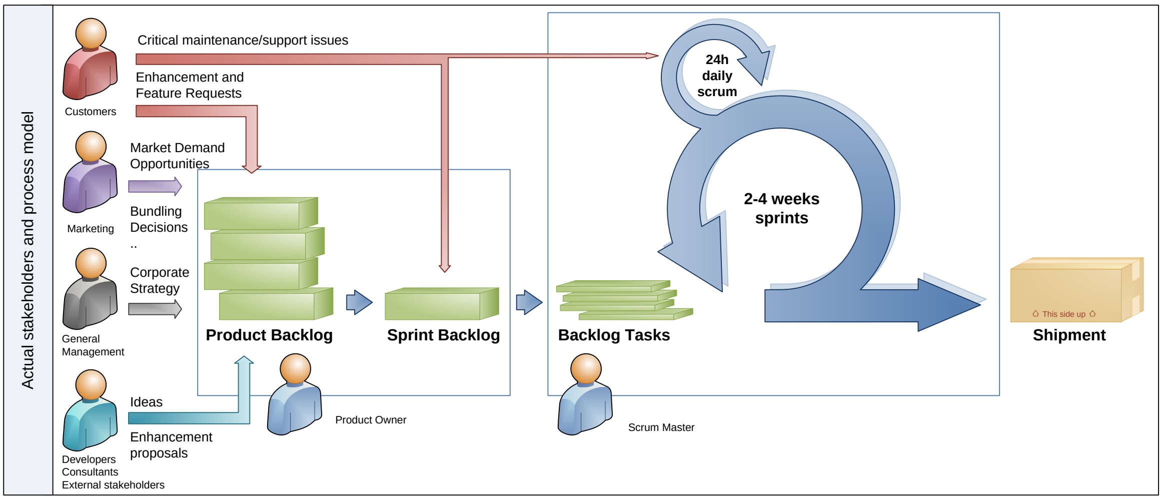


Figure 4 - Actual implementation of Scrum process

In the original Scrum model, the customer is the main source of input for the product owner. He defines the requirements, which are represented by items on the product backlog, ranked by the customers' priorities. While this might hold true for custom, project-based development, for the development of commercial, off-the-shelf (COTS) software, many more stakeholders are involved in practice and the product backlog is influenced by most of these (cf. Figure 4). Besides, in the agile teams we observed that not all activities are managed via the product and sprint backlog. Critical maintenance tasks are either directly added to the sprint backlog or the personal task list of developers.

3.1.4 Perceived weaknesses in the current model

From the analysis of the product roadmap decision making process, the decision criteria applied and the NPD process models, we identified several potential weaknesses in the current approach and confirmed their validity in interviews with relevant stakeholders. Table 2 summarizes these findings.

Weaknesses	Details
Roadmap priorities are not clear to all stakeholders.	There is no central and single source of information which transparently shows priorities from a roadmap perspective starting from a high-level perspective down to the implications on individual tasks.
Effort estimates are not precise (but are treated as such).	Effort estimates given during the initial, rough planning and scoping for a full release cycle are based on incomplete information available at this time. For instance, detailed functional and technical descriptions are created only after a feature has been approved on the roadmap. Yet, often initial effort estimates are treated as fixed and later deviations resulting from new information are seen as delays.

Market estimates and potential market impact forecasts are not precise.	In the area of market forecasts, rough generalizations and assumptions are made due to limited resources for a more detailed investigation. Yet, similarly to effort estimates, these numbers are used for goal-setting and treated as reliable. The process lacks a validity check for all types of estimates and forecasts.
Early fixedness on features limits agility and dynamic adjustment to emergent requirements.	The long planning horizon and the restrictive approval process for roadmap changes limit agility and often prevent upcoming opportunities to be pursued immediately despite clear benefits of doing so. The method implies a focus on executing what is planned rather than seeing the roadmap as means of orientation, which is yet subject to change if new information emerges.
On-costs are not transparent and remain unconsidered at planning time.	The whole roadmap planning process focuses on NPD costs from idea to launch. Yet, a significant share of product lifecycle costs accrues in the post-launch phase, and projects' post-launch costs significantly vary. A project might be favorable in terms of idea-to-launch costs, but not preferred due to its implications in the post-launch phase (e.g. by blocking valuable NPD resources).
Justification of technical enhancements and investments into platform technology is difficult based on classical financial planning tools.	With the current focus in the decision process (direct revenue potential, competitor analysis, and informal decision criteria) it is difficult to argue for important platform technology enhancements. Often, these do not yield any direct revenue potential and are also not visible in a competitor analysis, but are important enablers for other projects. There should be explicit room for these types of projects in order to provide a solid product portfolio over a long time period.
Deviations between planned vs. actually delivered roadmap are perceived negatively even if features have been abandoned in favor of more beneficial opportunities.	Due to its nature, the current process favors execution according to the plan over reaction to new opportunities, which might hinder innovation and striving for an optimum beyond the original roadmap plan. This view also makes it difficult for the product department to overachieve or deliver more than planned, other than by implementing the planned features in a shorter time frame. The consequence is rather conservative estimates in order to be able to achieve the individual goals.
Coordination across multiple teams with different development processes is difficult and not transparent.	With different development processes applied, there is a lack of global coordination and status sharing. In theory, the processes only deviate in how the design and implementation phases are structured. In practice, this means different tools and processes being applied and scattered data complicating global reporting.
Different understanding of flexibility among stakeholders on long-term planning vs. short-term agility.	There is no clear understanding of which time horizons are sensible to maximize short-term output while following a long-term vision. If market pressure increases, short-term projects which are supposed to show immediate results are preferred and long-term projects are abandoned.

No continuous involvement of all stakeholders.	Different stakeholders contribute in different lifecycle states, and are passive in others. This is natural for their activities and tasks, but the fact that there is no involved of certain stakeholders during long phases of the product lifecycle at all hinders a continuous decision process.
Business cases are not reassessed and continuously updated.	Market estimates and associated business cases for a project are created only during the initial roadmap planning and scoping process, but are not reassessed later, particularly after the launch to evaluate and learn from product success or failure and the accuracy of the planning process.
No ongoing decisions to optimize resource utilization.	Resource utilization is not transparent to all stakeholders. Thus there are no ongoing decisions on where resources will be spent best. This happens only in the beginning of the major release cycle.
Competing objectives of departments.	With the early fixedness and lack of flexibility in the planning process, departments tend to focus on their personal objectives rather than on achieving a globally ideal output. In concrete, departments focus on fulfilling their objectives to deliver the planned features within the estimated timeframes, but are not incentivized on seizing other opportunities. This might go as far as groups "booking" resources on the roadmap to generate leeway for their group at phases where no detailed planning is possible yet.

Table 2 - Perceived weaknesses in the context of roadmap planning

These weaknesses originate from a multitude of factors. Some are general communication issues, some are technical issues, but many of them stem directly from the roadmap planning and lifecycle management process. In the proposed approach we will address these.

3.2 Quantitative research results

3.2.1 Resource allocation and R&D efforts over the product lifecycle

Aside of the qualitative research discussed in the previous section, we intended to underpin these findings with quantitative data from R&D resource investments. In particular, historic data on logged NPD and maintenance efforts over the product lifecycle in period of 4 years has been analyzed. The analysis focused on the following questions:

- What is the ratio of effort invested in the individual phases of the application lifecycle, particularly in development and post-launch maintenance?
- Are there differences/patterns in the development of maintenance efforts over time for certain types of products and components?

- Is there a correlation between number of licenses sold and maintenance effort accrued over time, and if so, how is it characterized?

The overall objective was to understand how R&D resources are spent currently, and how accurately this is represented in the roadmap planning process.

The data analysis quickly revealed that given the current tools applied for data collection, overall efforts invested are not accurately represented in the data for continuous monitoring and planning of R&D activities. Yet, based on the accessible data and qualitative discussion of intermediate results with responsible stakeholders (managers of involved groups product management, product development, testing and quality assurance, product maintenance, customer support), the following high-level conclusions could be derived:

(1) 90% of enhancement projects consumed between 100% and 150% of the initial resource estimates at launch time. 10% consumed less than the initially estimated resource estimate until product launch.
(2) On-costs for post-launch maintenance vary significantly per enhancement and per project type whereas examples for both smooth and highly fluctuating maintenance effort curves have been found (see appendix section 7.2 for further details).
(3) Overall, post-launch maintenance effort accounts for 30-80% of total lifecycle costs, with a strong variance by project and component type. Overall, 48% of R&D resources are continuously invested in post-launch maintenance, based on a yearly average. This percentage increased over the past 4 years from 42% to 52%.
(4) There is not a significant correlation between maintenance efforts and number of licenses sold in the data. Yet, a number of stakeholders confirmed that there must be a clear positive correlation, with the maintenance effort increasing with the number of deployments.
(5) Requirements analysis and design phases have significantly longer cycle times with low total effort investment in relation to the development phase.
(6) During a fixed release cycle of 18 month, 30-40% of enhancements, or 15-20% of the total effort are added during the development phase and have not been considered in the initial planning.

3.2.2 Discussion of quantitative research findings

We discussed these findings with involved stakeholders to identify possible reasons and check their validity.
Actual project efforts mostly exceeded or meet the initial estimate, but are rarely lower (1). Given that estimation uncertainty should account for both over-and underestimation, this result indicates that dynamic scoping takes place, which was confirmed by development managers: If time is left in a project, it is invested to fulfill additional requirements that might have emerged during the implementation phase, or had minor priority in the initial planning.
The strong variance in maintenance efforts (2) potentially has multiple reasons, including support cases resulting from the release of dependent / interacting software components, peaks in projects around the end of sales quarters and the fiscal year, complexity of the component, quality of the

documentation, initial effort invested in testing and quality assurance and more. Still, certain classes of maintenance efforts can be defined and used as a reference model during planning. This aspect is discussed in greater detail in section 4.5.1.2. The reference model is discussed in the appendix, section 7.2.

Maintenance is a significant driver of total development costs (3), which is also confirmed in the literature (see April and Abran [4]). Maintenance has been split into separate teams in order to shield NPD resources from these activities. Among other positive and negative effects of this step one of the main results is that maintenance efforts are more or less fixed with the number of resources available to accomplish these tasks, which is why maintenance efforts are not currently considered in the roadmap planning process. However, decisions at roadmap level heavily impact later maintenance efforts. Thus, the proposed roadmap planning model is based on total product lifecycle costs rather than idea-to-launch costs.

The long cycle times during the concept creation, requirements analysis and design phases (5) might originate from how these functions are integrated into the overall NPD process. Compared to implementation and testing tasks which are clearly structured, prioritized and scheduled, for these activities there are often multiple stakeholders involved (product design, a solution architect and development managers), which lengthens the process. The planning activities and design by themselves are not precisely estimated, their dependencies are often overlooked and the central role of these tasks is not sufficiently accounted for in the process. Out of these considerations, we propose to treat these tasks more explicitly, and rank activities separately in different lifecycle stages. Furthermore, we integrate value-stream mapping as a tool to identify process bottlenecks.

The fact that 30%-40% of enhancements are added after the initial scoping (6) confirms the need for "planned flexibility" in the process. An analysis of such enhancements shows that all of these are valid and either absolutely necessary from the users' perspective, or very beneficial opportunities. Yet, the problem with the current approach is that mostly projects are added to the roadmap, but no equivalent of projects in terms of required resources is removed. The objective for the proposed model is therefore to enable a transparent "in-and-out" process.

Abstract. *Ranking is a powerful tool to enforce, communicate and execute clearly defined priorities. The proposed framework is built upon the idea of establishing and constantly updating a ranked list of projects on the roadmap. Different decision criteria are analyzed and discussed, from financial valuation to the users' perspective. A set of evaluations and roadmap balancing techniques are applied to aggregate all criteria to a consistent ranking. This ranked list is the basis to schedule activities in operations. Innovation controlling metrics are used to monitor the execution and refine the decision process.*

4 AGILE PRODUCT LIFECYCLE MANAGEMENT FRAMEWORK

Based on the empirical findings and observations discussed in the previous section, a product lifecycle management framework is proposed in order to support mastering the challenges in balancing innovation and continuous improvement in complex products' lifecycles. It is grounded on the ideas of the resource-based view on a firm (theory originating from Pfeffer and Salancik [67]). Based on all known and relevant decision parameters, the model aims to dedicate available resources to the most critical and important tasks at any point in time. These include all tasks directly related to the product development and market introduction, but also investments in information gathering in order to reduce technology and market uncertainties as well as exploration and research activities. Thus, it requires continuous decision making at different granularities and a structural alignment of decisions and underlying operations based on a seamless information flow.

4.1 An alternative view on NPD and its context

In contrast to existing NPD models as discussed in section 2 (e.g. activity-stage models such as the Booz, Allen and Hamilton model of new product development or the traditional Stage-Gate© approach), the model proposed in this thesis attempts to extend the view on NPD by focusing not only on one process instance from idea to launch, but a constant stream of parallel activities, i.e., ideas, projects or products in different stages of their lifecycle that need to be managed, including the products in the post-launch phase. Ultimately, there is a limited pool of resources which needs to be allocated between these projects.

Hence, optimizing time-to-market and the NPD process for one project is not sufficient if parallel or dependent projects are not managed and other opportunities remain unconsidered. Table 3 compares characteristics and focus of existing NPD literature with the proposed approach. Figure 5 illustrates the proposed view on NPD as a constant stream of activities from the fuzzy front-end to structured NPD activities to post-launch maintenance and support.

Existing Literature	Proposed Approach
NPD as a (linear) process from ideas to launch, with intermediate selection and decision processes (e.g. traditional Stage-Gate© model [22]).	Development as a continuum with regular inflow (ideas) and outflows (products) and constant decision processes.
Focus on lifecycle of one product.	Focus on resource utilization for complete product pipeline.
Optimize idea to launch process.	Optimize resource utilization over the complete product lifecycle.
Separation of product development and post-launch maintenance activities.	Dynamic pool of resources allocated to product lifecycle activities as required.
Restrictive go/kill decisions.	Option creation and evaluation; Targeting resources to most beneficial projects.
Isolated view on projects: Fixed scope, plan vs. actual implementation monitoring.	Fixed resource investment, dynamic scope, and flexible adjustment to emergent requirements.
Projects funnel.	Projects stream.
Focus on NPD costs.	Focus on total lifecycle costs.
Projects competing for resources and sponsorship.	Constant seek for *global* ROI optimum.

Table 3 – Comparison of existing literature and proposed approach: an alternative view on NPD

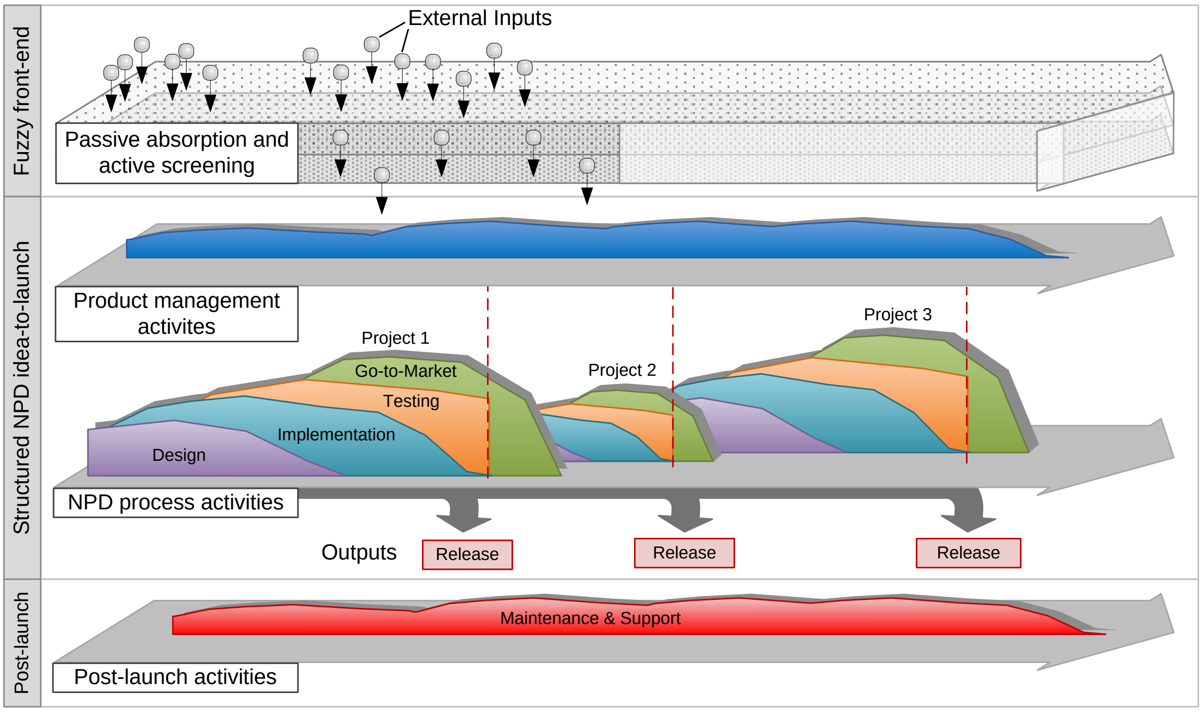


Figure 5 - NPD as a stream of projects and activities

This alternative view on NPD poses new challenges on product lifecycle management. Decisions have to be carried out in a much broader context of all impacted resources and dependent projects, planning needs to consider activities and resource needs beyond the launch phase, Closer interaction and constant coordination between all stakeholders is required to balance all parallel activities. At the same time, the potential benefit is undeniable: Achieving a global optimum of returns on R&D investments.

Throughout the subsequent sections we present a product lifecycle management framework for implementing this view on NPD. We start by discussing the general requirements and objectives for such a framework.

4.2 Model requirements

Literature and mainly the experience with existing models have shown that in order to be successfully implemented and executed, a product lifecycle management model needs to fulfill several basic requirements, independent from its actual methodology.

Requirement	Explanation
Transparency	When it comes to optimization methodologies, algorithmic/mathematical solutions exist which are able to compute an ideal or close to ideal solution for a decision problem given the provided input data. While powerful, these methods often have the character of a black-box solution and suffer from a lack of transparency over the ultimate decision process. It is not necessarily clear to stakeholders, why a certain solution has been chosen over others. In contrast, the proposed model shall be transparent in order for decisions to be easily communicated and reasoned to all involved parties.
Simplicity first, complexity on demand	The decision problem over a product roadmap is inherently complex. It is a defined objective of this model, to reduce complexity for the decision maker rather than introducing further complexity by increasing the number of decision parameters and models.
Flexibility	Every product lifecycle is different, involved people and existing processes are different. The model shall offer sufficient flexibility to be tailored to a particular application without losing its benefits.
Conformance and Integration	The model shall integrate seamlessly with existing processes, e.g. in product development. It shall be an intelligent layer on top, integrating the roadmap planning and lifecycle decisions with underlying operations without being too disruptive.
Light-weightiness	The model shall not introduce bureaucracy, complex roles, responsibilities and processes requiring dedicating valuable resource load to the implementation of the process itself.

Table 4 - General requirements for a product lifecycle management framework

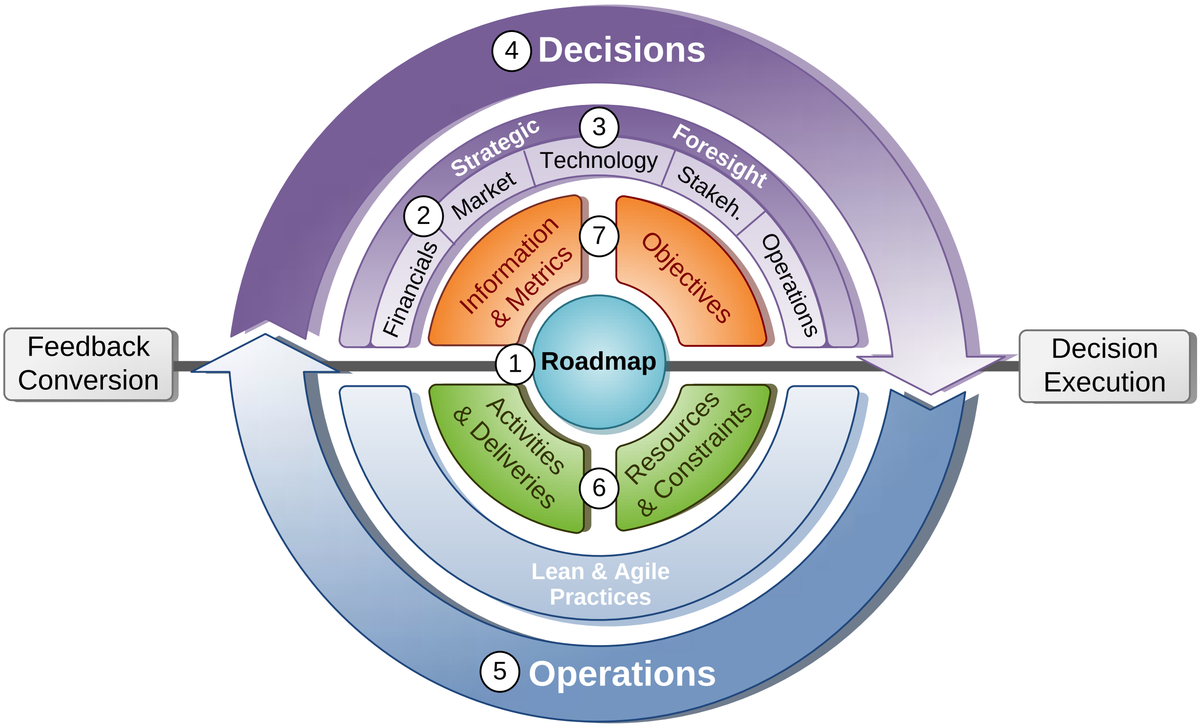
4.3 Model objectives

In the introduction, the high-level objective this thesis has been described as a framework for product lifecycle management under uncertainty. The field study as well as the literature review pointed out concrete challenges and shortcomings of existing approaches. Based on these, the list of objectives for the proposed model has been further refined in order to evaluate its effectiveness in a later step. The model shall be capable of (1) enabling continuous product lifecycle and technology roadmap decisions by gathering the relevant information and providing a decision and prioritization model. (2) Incorporate principles of flexible and agile product development into plan-driven approaches in order to align sustained engineering with the flexibility to react to emergent requirements and information and leave room for experimentation. (3) Enable transparent decisions and impact analysis for potential roadmap decisions. (4) Reduce the cycle time from ideas to launch. (5) Provide a holistic view on the product lifecycle, in particular on post-launch maintenance and costs. (6) Alleviate early fixedness and restrictive go/kill decisions based on incomplete information and (7) guarantee resource investments in must-have features as well as avoid resource investments in irrelevant tasks.

4.4 Model overview

The proposed product lifecycle management framework builds upon establishing and maintaining a roadmap defined as a ranked list of enhancement projects. The concept of ranking is suggested for several reasons: First, it enforces a clear commitment to priorities from involved stakeholders. Second, it provides an ideal input for operations by defining in which order activities shall be performed according to their priority. Third, it is easy to present and communicate.

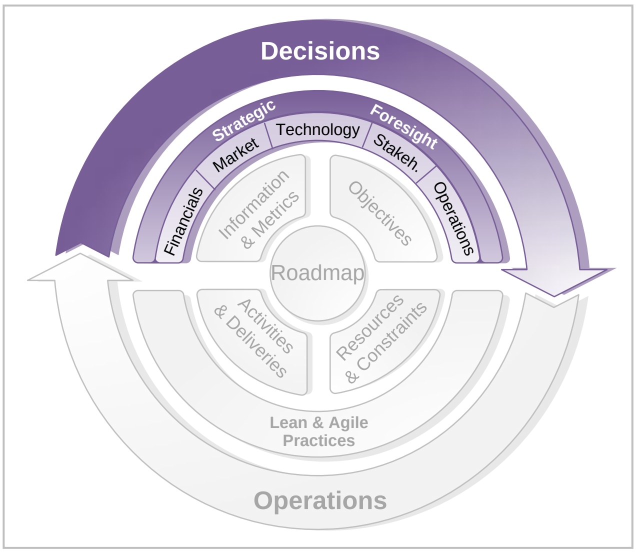
The model focuses on two key aspects and their interplay: **Decision making and prioritization** is required in order to establish and continuously update the projects' ranking on the roadmap. Different perspectives on the decision process are discussed, including financial valuation, the market perspective, stakeholders' needs, the technology and the operations perspective. Relevant decision criteria from these perspectives are aggregated to establish a final ranking. For this purpose, an informal, team-based approach, a semi-structured method and a fully structured decision analysis approach are presented. Depending on the complexity of decisions and the available time, the suitable approach can be selected. Further relevant aspects to decision making are to balance the portfolio of projects based on risks, timing of market impact and other factors, and innovation controlling metrics which measure decision results and allow refining decision priorities. **Operations** ensure the efficient execution based on roadmap priorities and resource constraints and dependencies. Resources are targeted towards the most impactful activities. The process deals with inherent uncertainties and keeps utmost flexibility for changes. We discuss resource planning and roadmap scheduling, as well as how to deal with continuous roadmap changes. The operational process model defines which roles and stakeholders contribute to which stages of the product lifecycle.
Figure 6 summarizes the core building blocks of the framework. Throughout the subsequent sections, these steps are discussed along with proposed techniques and frameworks to assist a practical implementation.



1	The roadmap is managed as a ranked list of enhancement projects. The roadmap is shared transparently with all involved stakeholders.	
2	Different decision perspectives are analyzed for performing the projects ranking.	*see section 4.5*
3	The roadmap is the melting pot between strategic foresight activities and current operations.	*see section 4.5.6*
4	The decision process aggregates the perspectives into the final ranking and keeps a balanced portfolio of projects.	*see sections 4.5.7 and 4.5.8*
5	Based on the ranking, operations are aligned with all continuous roadmap decisions and immediately adapted to changes.	*see section 4.6*
6	Activities are scheduled to maximize the output with respect to roadmap priorities and available resources and constraints.	*see sections 4.6.3 to 4.6.6*
7	Innovation controlling metrics are used to measure results and adjust priorities in the decision process.	*see section 4.7*

Figure 6 - Overview on product lifecycle management model building blocks

4.5 Decision making and prioritization



In contrast to existing portfolio optimization approaches (see section 2.1), the objective of this work is to abandon the idea of a closed algorithmic solution to the optimization problem in favor of a transparent selection process, where the final output is intuitive and the process can be demonstrated and explained to all stakeholders.

First, it is necessary to understand the different perspectives from which product roadmap decisions can be analyzed. The **financial perspective** attempts to estimate the return on investment (ROI) of a potential project. The **market perspective** is closely associated with the financial perspective in terms of customers' willingness to purchase a product, but analyzes customer needs from an abstract perspective and also takes into consideration other market players' activities and offers. The **stakeholders' perspective** captures the needs and interests of internal and external stakeholders, particularly employees, partners and shareholders. The **technology perspective** takes into consideration short-term and long-term technology trends and advances which might be exploited or might threaten a firm. The **operations perspective** considers resource availability and constraints, as well as the impact of decisions on resource utilization during planning. We claim that all of these perspectives have to be taken into consideration and when making continuous decisions along the product lifecycle, and each requires different tools and methodologies.

4.5.1 The financial perspective

The field study on current roadmap decision criteria (see section 3.1.2) has confirmed that the financial perspective is in practice one of the main decision drivers for product roadmap decisions despite the shortcomings of common practices such as NPV (net present value) analysis [15][19]. First, uncertainties and flexibility in the execution are not valued. Agile operations allow reacting to emergent requirements or upcoming opportunities. Certain projects might provide later options for being leveraged in a different context, others may not. This is not accounted for in NPV analysis. Uncertainties are considered only in the sense of best case, average case and worst case analysis. Options to react to a worst-case scenario are not considered again. Second, total lifecycle costs are often not considered by just looking at NPD costs from idea to launch. NPV analysis, from a conceptual perspective, is capable of incorporating on-costs into the financial planning based on expenditures over the planning period. Yet, in practice we observed that these either remain totally unconsidered or are based on rough and undifferentiated estimates, whereas on-costs might strongly vary per project or per project category.
In order to address these aspects, three tools are proposed next to traditional NPV analysis and

planning: the integration of options thinking into the planning process, an on-costs reference model and the application of target costing and scoping.

4.5.1.1 Integrating options thinking

Literature on corporate financial management presents real-options analysis as an alternative to using DCF (discounted cash flows) analysis, in order to account for the fact that projects are managed actively, and managers can take advantage of upcoming opportunities or new information. For example, instead of running straight into the worst-case scenario identified in the initial planning, a manager might abandon a project or at least shrink it. If a project develops positively, a manager might even expand it, to increase profits beyond the initially assumed best-case scenario. In literature, options to expand, to wait and learn before investing, to shrink or abandon a project, or to vary the mix of output or the firm's production method have been discussed [15].

Different models and methods have been proposed to deal with real-options valuation, including partial differential equations (PDE), dynamic programming, decision trees, and Monte-Carlo simulation. The most well-known method is the Black-Scholes Formula, originally proposed by Black and Scholes [11] for the valuation of financial options. An example for the valuation of options in a software project based on this formula is provided in appendix section 7.3. Out of the set of current models, we believe it is the best choice for valuing above-said options. In practice, however, we believe that this model's applicability is nevertheless limited due several implicit assumptions: (1) all uncertainties are assumed to be resolved when deciding to exercise an option, which is in practical scenarios never the case. There is always remaining uncertainty. (2) It assumes a single exercise date as common with financial options. (3) It assumes that the uncertainly in a project's cash flow follows a log-normal distribution. In contrast, in many practical scenarios project success or failure is rather close to a binary distribution.

Numerous discussions around the applicability of real options for R&D valuation exist. Damodaran [28] argues that options are ubiquitous to any business decision, but it is necessary to clearly distinguish those with significant, quantifiable value. Brealey et al. [15] acknowledge these challenges as well, but as a common ground in the discussions underline the *qualitative value* of the approach. By raising awareness on alternative options during the decision process, a DCF-based financial valuation might be challenged. This position is also supported in this thesis. Thus, the integration of real-options thinking does not rely exclusively on hard numbers but is integrated as a decision making tool to provide a more differentiated view on financial valuations. Table 5 provides examples for different option types from the software domain.

Option	Example
Option to expand	**Example 1.** Implementing feature A now enables the integration of tool B later to open a new market.
Option to gather more information before deciding	**Example 1.** Feature B might have strong market impact, but requires intensive resource investment. If we do feature C and give it to customers, we will get feedback on the potential value of not only C,

	but also feature B, so that we can decide it is worth the effort. Thus, C has an option value for B. **Example 2.** Feature D's technology is highly uncertain. Yet, for feature D there is a parallel research project, which will demonstrate the technology's capabilities in approx. 6 months from now.
Option to abandon a project early	**Example 1.** If we do feature A, it gives us early feedback and we could also abandon it if negative, compared to feature D, where we do have a big investment before getting any feedback/results.
Option to recombine/reconfigure the outcome	**Example 1.** If we do feature A, it is of general interest and could serve customer group X, but could be alternated to serve customer group Z in a different context as well.

Table 5 – Real options in product roadmap decisions

Calculations based on sample projects (an example is provided in Appendix section 7.3) carried out in the course of this work have confirmed that for these examples of options, the financial valuation may be significantly different whether real options are considered or not. At the same time it became apparent, that the valuation model is very sensitive to all numeric parameters, such as the assumed volatility and distribution of uncertainties. Therefore, the following approach is proposed when determining the best financial valuation among a set of candidate projects:

1. Compute projects´ valuation based on DCF and rank them by value.
2. Identify significant options associated with each project (we propose a set of key questions for orientation and discovery of these questions, see Listing 2).
3. Select the options that are immediately quantifiable in financial terms.
4. Compute the options value based on Black-Scholes model and add to net present value.
5. Re-rank the projects.

Listing 1 – Process for applying options valuation in roadmap planning

Listing 2 summarizes few key questions for identifying options associated with a project.

- Does this project contribute common platform features which can be reused later and are valuable in other projects?
- How quickly can we get feedback on potential failure/success?
- Can we reduce uncertainties before starting the project?
- Can this project be expanded later or does it have a fixed maximum scope?
- Does this project enable further opportunities known at this point in time?
- How specific/generalizable is this project? Can it be reused outside the core need fulfilled by it?

Listing 2 – Key analysis questions for identifying risks and options

In case no significant, financially quantifiable options can be identified, non-quantifiable options can be considered as indicators for planning flexibility. The more basic options can be identified, the more preferable should a project be, assuming that planning flexibility can always be exploited.

4.5.1.2 On-cost estimates based on a reference model

In order to account for the total product lifecycle costs in the roadmap planning process, post-launch maintenance efforts need to be considered aside of NPD costs at the time of planning. Technically, this aspect is not challenging. Resource needs for post-launch maintenance have to be estimated for a project, and their costs can then be computed. However, in practice this procedure is potentially time-consuming, error-prone, and there is large variance in the maintenance efforts.

In order to simplify the planning process, we propose using a reference model for maintenance efforts based on industry-average figures or historic maintenance efforts. We identified two relevant dimensions along which to classify projects' maintenance efforts: The total maintenance effort over time and the correlation of maintenance effort with the number of product items sold. The first dimension describes the maintenance curve based on a fixed number of product items in use at a fixed number of customers, whereas the second dimension describes the changes of this curve in relation to the quantity of product items in use.

Appendix section 7.2 discusses different maintenance effort patterns observed in history records, ranging from constant maintenance effort to irregularly varying effort to initial peaks after product launch and a subsequent steady decrease. Based on analogous projects or a project's characteristics, the most suitable reference model can be selected and considered for the post-launch resource planning. Relevant characteristics for software projects include their dependency/integration with external systems, scope and complexity of the user interface, diversity of application scenarios, frequency of use, as well as skills and characteristics of the target user group.

4.5.1.3 Applying target costing and scoping

In the traditional approach towards effort estimates, tasks are performed in the following order: (1) create rough estimates on a project in order to define an overall scope, (2) write product specification and do detailed planning, (3) make detailed estimates based on the specification, (4) perform resource planning and derive realistic target delivery date. This approach is preferable in cases where the project scope is fixed and invariable, and little to no change is expected during the implementation. In contrast, agile practices suggest the use of target scoping, i.e., not time becomes the variable dimension, but the products' feature scope. In this approach, first a rough estimate is done to determine a target delivery date. Then the product is created in short-lived iterations to the extent possible in the estimated overall time frame. The scope might be extended or narrowed if actual efforts were incorrect or emergent requirements are favored over originally planned features. This approach is preferable for delivering frequently and regularly, and it implicitly deals with planning uncertainty by adjusting along the way. Yet, applied carelessly there is a danger of scoping out important product features which are necessarily required for the product to be successful. In marketing literature, the related concept of target costing is proposed (see Horvath [43]), which focuses on the question of "how much are we allowed to spend on the product" based on a market perspective.

We propose a hybrid approach towards target scoping and target costing based on a distinction of different product features. For this distinction, we apply the Kano model [47] (see Figure 8). For features which provide a product's *basic attributes*, the minimum deliverable feature set is identified, specified and estimated according to the traditional, plan-driven approach. Here, the scope is not variable afterwards and a delay in implementation requires delaying the product launch. For features providing a product's *performance attributes*, similarly the minimum deliverable features set is identified and fixed initially. Other performance attributes may be added based on a targeted resource investment scope, as done in agile approaches. Features providing *excitement attributes* are not fixed beforehand, but are only implemented within fixed target costs. The more excitement attributes can be delivered within the target scope of resource investment, the better. The process is illustrated in Figure 7.

In practice: Combining predictive and adaptive planning

In practice, this combination of estimates for the technical minimum scope and target costing has significant impact on the way the roadmap development is managed. It implies that beyond the basic attributes, there is a flexible and dynamic buffer (up to the target scope) which can be filled incrementally with the most beneficial features. The development department contributes the estimates the minimum scope and starts the implementation with the initial technology set. Product management and marketing has planning flexibility and leeway for late decisions to focus on particular performance and excitement attributes according to emergent requirements.

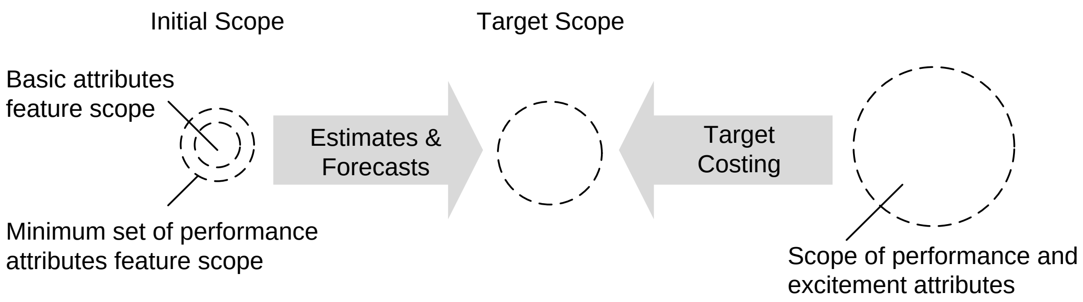


Figure 7 - Target costing and scoping process

This mechanism is the basis for balancing plan-driven and agile methodologies and combining the advantages of both paradigms.

4.5.2 The market perspective

Implicitly, the financial analysis of products assumes a certain value generation for the customer, which he is willing to pay for. The market perspective attempts to identify new opportunities for customer value creation, and to maximize the value generation with the delivered products by matching them ideally with customers' needs. Thus, deciding on projects and product enhancements from a market perspective means assessing which candidate project is most likely to generate the maximum value for customers which they are willing to pay for (directly or indirectly). In order to perform this decision, we identified the two key aspects. First, the product and its features have to be oriented towards users and their needs. Therefore, we suggest adopting and integrating methods of user-oriented design into the framework. Second, it is necessary to decide when a product is mature and saturated in a sense that further enhancements generate little customer value and it is better to invest resources into new product lines. Both aspects are subsequently discussed.

4.5.2.1 Integrating methods for user-oriented design

In marketing and innovation management literature, different models for user-oriented design have been proposed, including empathic design [53], the Kano model [47], conjoint analysis [35], or quality function deployment (QFD) [1]. Empathic design focuses explicitly on the design process, to understand latent needs and requirements which are not directly and explicitly expressed by stakeholders, or are intangible. Concepts of empathic design can be implemented in activities during the concept creation phase or in exploratory research. Yet, unlike other methods it does not provide direct input to the roadmap decision making process.

The Kano model has been discussed in the previous section as a tool for distinguishing between different feature categories in the planning and target scoping process. Figure 8 illustrates the model's core contribution, the distinction of customer preference into different categories.

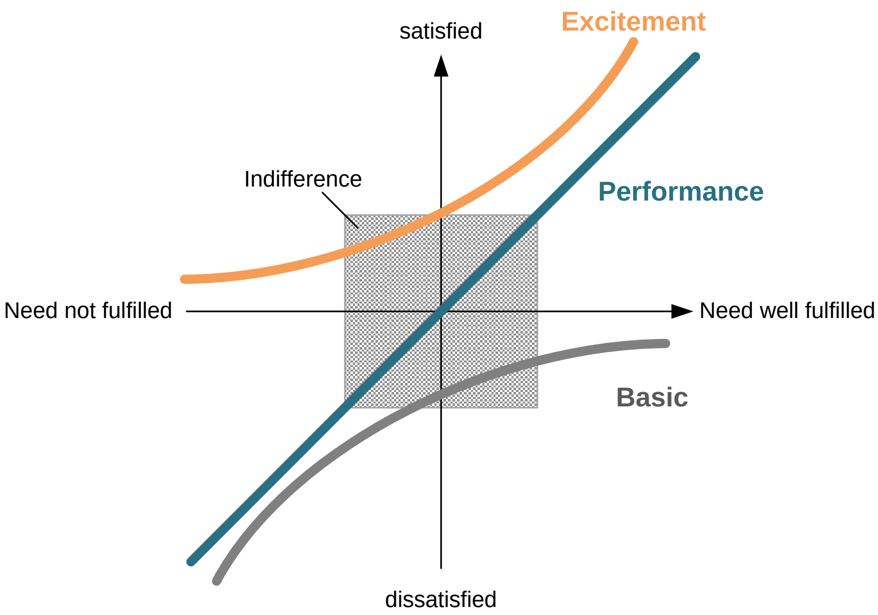


Figure 8 - Kano model

Source: Kano, 1984 [47]

This model can be used not only in the scoping and project valuation model, but also to make roadmap decisions from a market perspective. By analyzing a project's estimated ratio of required basic attributes to the total effort or identifying whether and with which effort excitement attributes can be added, this tool can be used for establishing a ranking of projects. Listing 3 summarizes several key questions for carrying out this analysis.

- What is the approximate ratio of basic attributes we need to deliver to be successful with this project? (The lower this number, the more relative impact will resource investments have as they contribute to performance and excitement attributes.)
- Can this project deliver excitement attributes, and if yes, how much is possible in the defined scope? Whom will this project excite and why?
- In terms of performance attributes, if we create this product, how will it perform technically relative to competitive products?
- Can the product performance be up-scaled in later versions based on the implementation strategy?
- Is what we deliver with a project clearly outside the space of indifference when looking at the current market offering?

Listing 3 – Key analysis questions for applying the Kano model to roadmap decisions

Answering these questions is supported by the basic Kano model analysis approach, which combines a functional question on a feature (rating of satisfaction if a product has a feature), with a dysfunctional question (rating of satisfaction if a product does *not* include a feature). In this way, the core features of a project might be analyzed to answer above-said questions, particularly if stakeholders disagree on the answers.

In addition to the Kano model for decision making from a market perspective, we propose applying core concepts of QFD. In brief, QFD captures different “voices” of customers, engineers, and the competitive position. The method analyses dependencies between technical quality elements and customer requirements. Intuitively, the more a technical quality coming with a product enhancement contributes to different and relevant customer requirements, the more preferable it should be in the product roadmap planning.

We propose applying a simplified subset of the original QFD model, which focuses on product features and quality attributes. The model is illustrated in Table 6. First, relevant product quality attributes and features are identified and a comparison to competitors is done. This comparison is the main determinant for a product quality’s relative weight and importance. Then, projects are compared with respect to their contribution to establish or enhance the individual product qualities. The aggregation of these values establishes an overall ranking of projects.

Competitive comparison in product qualities

Product quality attributes	Our product	Competitor A	Competitor B	Competitor C	Competitor D
PQA 1	10	6	4	8	9
PQA 2	2	7	5	10	5
PQA 3	6	10	7	6	8
…					

Projects' contribution to product qualities

Product quality attributes	Weighting	Project A	Project B	Project C	Project D	…
PQA 1	1	4		10	1	
PQA 2	10		3	1	7	
PQA 3	7	1			2	
…	..					
Weighted total		11	30	20	**85**	..

Scale: 1 (lowest) – 10 (highest)

Table 6 - Simplified QFD model for roadmap decision making

4.5.2.2 Switching resources to new products

Aside of targeting product enhancements and projects best to customers' needs and discovering areas with highest possible market impact and customer value generation, another important question in product roadmap decisions is whether and how many resources to invest in existing projects and how much to invest in new products. From a theory perspective, the technology S-curve model attempts to answer this question. It originates from the early work on innovation by Schumpeter [79], and has been applied and popularized amongst others by Christensen in his analysis on disruptive technologies [18]. Christensen and other authors such as Utterback and Acee [92][93] or Tushman and Anderson [91] support the view that technologies' performance follows an S-shaped curve along their lifecycle, until they are overtaken by a superior, disruptive technology (see Figure 9).

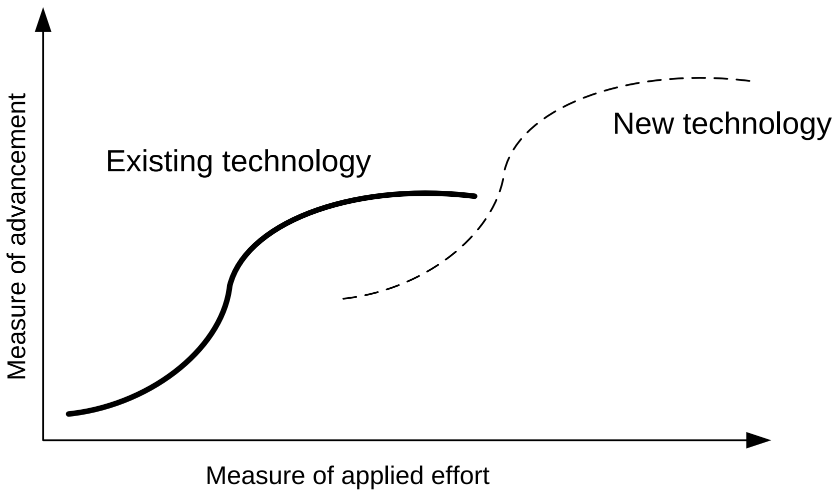


Figure 9 - S-curve of technology advancement

Source: Adapted from Christensen, 1997 [18].

This theory has gained great popularity since its publication, but raised several critics as well. Amongst other authors, Sood and Tellis [87] recently argued that empirical findings do not generally support the S-curve theory and show examples of technologies that appeared to have reached the plateau phase, but have advanced significantly again after a period of time, or potentially disruptive technologies which have never overtaken the existing technology.

We believe that the basic S-curve model has its validity, but depends strongly on the metrics being applied to determine the measure of product advancement. Thus, we propose analyzing and identifying relevant indicators for a particular product to determine product maturity and product potential for advancement. Our case study has shown that these indicators may vary strongly based on the type of product, distribution, sales and deployment structure. Both previously discussed approaches, the Kano model and QFD can be applied equally in this context. The Kano model supports the identification and assessment of potential for excitement attributes and the upper bound for performance attributes. QFD judges product maturity relative to competitors. Further relevant indicators for product quality and maturity as well as market potential can be obtained from the appendix, section 7.4.

4.5.3 The stakeholders' perspective

Both the financial and the market perspective on roadmap decisions are undoubtedly the predominant decision factors documented in literature and used in practice. Yet, we want to point out other aspects as well which may implicitly impact the decision process in practice, and which shall be included in a structured decision process. Among these is the stakeholders' perspective, i.e., the interests of all stakeholders in the NPD process, the product launch and later product use, providers of complementary products and services, etc. Each aspect in itself may require a thorough and detailed analysis, which is well beyond the scope of this thesis. Therefore we propose a pragmatic approach to include the stakeholders' perspective into the decision process and to include specific analysis techniques on demand to refine results: We developed a checklist of key questions for judging benefits and potentially negative consequences for various stakeholders. Adjusted to a specific organization's characteristics, this presents a valuable qualitative tool to be aware of stakeholder interests in the decision process. Only if conflicts of interest are discovered, these should be illuminated in greater detail. Examples for relevant questions developed in the course of this work are summarized in Listing 4. We raise no claims to completeness. Rather, this should serve as an initial orientation for developing a checklist matching an organizations' individual situation.

- Does the product/product enhancement satisfy the interests of all people along the value chain, including the final users, the decision makers on the purchase (if different), sales people who demonstrate it, administrators of the software, consultants, and implementation partners?
- Does the project impact complementary services provided by partners?
- Does the project impact solutions and extensions created by customers, or generally the current usage scenario?
- Is the project attractive and understandable for investors, the board of directors and top management?
- Is the project attractive for employees to be built? Does it have a motivating and inspiring vision?
- Does the project allow for new integrations and add-ons created by customers or partners?
- Can the product benefit be demonstrated easily to existing and new customers? What does it take to demonstrate its value?
- Can the project be used and applied internally to get real-life feedback before exposing it to the public?

Listing 4 – Key analysis questions for the stakeholders' perspective

4.5.4 The technology perspective

The technology perspective on roadmap decisions is concerned with *how* a certain project will be implemented. The technology perspective is to a higher degree industry specific as compared to other decision criteria. Relevant aspects in the software domain are industry standards and technology platforms (e.g. operating systems, programming languages and runtimes, server technologies, communication protocols, integration methods, etc.), technology trends (e.g. virtualization and cloud computing, web applications, etc.) and technology leverage (e.g. interoperability, extendibility, leveraging external sources such as open-source projects). Technology roadmaps of other vendors, particularly the providers of underlying technology and systems, are a relevant source in this context.

During the roadmap planning process, viable technology alternatives for a product need to be identified. These alternatives typically will show different characteristics in terms of product qualities, resource needs and costs for development. Based on these characteristics, a preference ranking from a technology perspective is derived.

4.5.5 The operations perspective

Finally, the operations perspective analyzes roadmap decisions from a resource and process perspective. Which resources does a project require? Are these resources available, or do they have to be acquired? Are there significant overlaps in resource needs among projects? We will illuminate this aspect separately in section 4.6 on operations. The model is designed to have a current view on resources in parallel to making roadmap decisions and prioritizations. Depending on which priorities are set, the model points out the implications on current resources (shortages or slack capacity).

4.5.6 Strategic foresight and the technology roadmap

Well documented in literature and underlined by observations in the course of our research, one of the most challenging aspects in product roadmap planning is accounting for and preparing towards long-term evolutions of markets and technologies. Bold moves in the marketplace typically require significant R&D investments, while at the same time they are associated with high uncertainty. The tradeoff between harvesting short-term, low-hanging fruits and pursuing long-term goals reflects in the allocation of resources on the product roadmap. In practice, if not actively supported and promoted, there is a tendency to cut resources in exactly these long-term endeavors if short-term projects lack resources. This tendency towards neglecting long-term developments has been researched amongst others by Levinthal and March in their work on the “myopia of learning” [54] in the context of organizational learning. From a manager’s perspective, it is much easier to defend a delay in a visionary, high-uncertainty project rather than a short-term operation planned in detail beforehand. In large firms, visionary projects are often pursued in separate research departments. This ensures the resource commitments, but at the same time the transition to regular product development requires complex interfaces between the department silos. Smaller firms often cannot bear the costs of a separate research unit and visionary projects have to share resources with regular product development. The subsequent discussion refers to this scenario.

Out of the above considerations, we see two relevant aspects in the market foresight perspective on decision making: First, long-term visions and opportunities have to be identified and pursued in concrete R&D projects. Second, resource investments in these projects must be well defended and reasoned in order not to be postponed in favor of short-term efforts. An extensive body of knowledge exists in the area of corporate foresight. Daheim and Uerz [26] [27] name four dominant paradigms in the past five decades: expert-based foresight, model-based foresight, trend-based foresight and most recently context-based, open foresight and argue that nowadays, trend-based and context-based methods are predominant in practice. Trend-based foresight attempts to project developments based on indicators for large trends. The early stages of such trends are identified by so-called weak signals [3][16]. Context-based methods focus on understanding and anticipating, but also shaping change.

Vecchiato and Roveda [94] provide a general classification of foresight activities (see Figure 10) and summarize models applied in practice and discussed in literature along the proposed classification schema. Across industries, monitoring and scanning, Delphi analysis and scenario analysis are common techniques in corporate foresight operations.

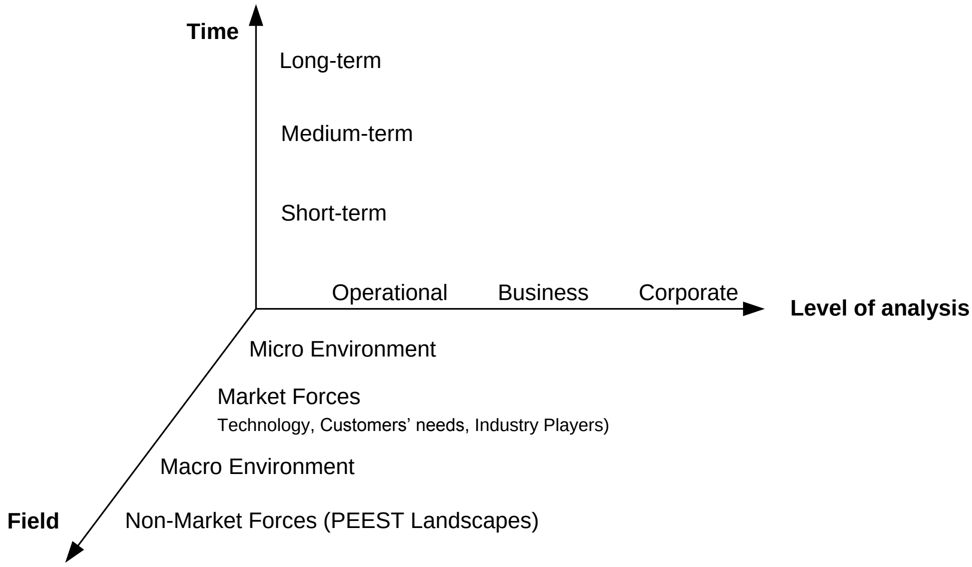


Figure 10 - Classification of strategic foresight activities

Source: Vecchiato and Roveda, 2010 [94]

We identified a forecasting method developed by Siemens AG named "pictures of the future" [85] as a particularly valuable concept. It combines extrapolation from the current business, products and technologies with retropolation from business scenario analysis. The concept is illustrated in Figure 11.

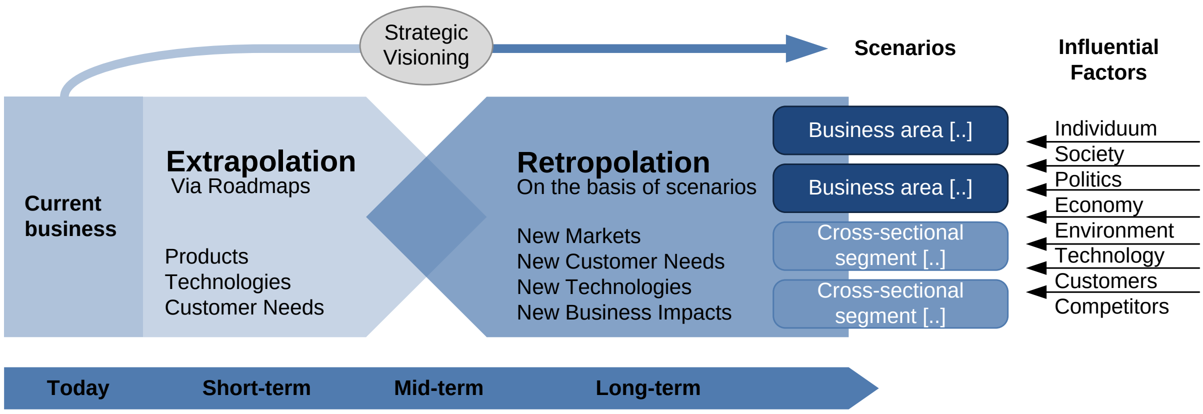


Figure 11 - Picture of the future: corporate foresight method

Source: Siemens AG, 2005 [85]

While foresight methods have been well researched and will not be discussed in further detail in this thesis (the interested reader may refer to Hines [42] for an analysis of state of the art in strategic foresight methods), both the organizational anchoring and the process design of foresight activities have rarely been discussed. Rohrbeck and Gemünden [73] identified three roles for corporate foresight in order to have significant impact on the innovation capacity of a firm: *"(1) the strategist role, which explores new business fields; (2) the initiator role, which increases the number of innovation concepts and ideas; and (3) the opponent role, which challenges innovation projects to increase the quality of their output."* We claim that the melting pot between corporate foresight and regular operations is exactly the product roadmap, and different stakeholders involved in roadmap decision making may take either of these roles implicitly. The "pictures of the future"-method integrates seamlessly with this concept: The product roadmap is extrapolated from current projects to bold market and technology moves based on scenario retropolation. The further ahead, the less detailed the roadmap and the more significant the changes compared to current operations. This concept has significant impact on the organizational anchoring of foresight: Being aligned with the product roadmap, product management becomes the major driver and consumer of foresight activities. Results reflect and are incorporated into product advances and with that rolled out to the entire corporation.

In practice: Linking strategic foresight to daily business

In practice, despite the existence of advanced methods for strategic foresight we observed that a major challenge is to derive concrete product enhancement proposals from the output of foresight activities such as identified scenarios. We claim that despite established tools and processes, it requires individual creativity to bring the abstract level of scenarios to a concrete level of product enhancements. In section 4.6.9.3 we discuss informal roles which drive this creative process. We recommend for a practical implementation to identify people which potentially match these roles (e.g., technology or market visionaries, opponents and challengers) and involve them with the conversion of foresight results to projects in individual and team activities enabling their potential.

The second aspect of securing R&D investments into long-term, visionary activities will be discussed below in section 4.5.8, "Balancing the product portfolio". For roadmap decisions, we propose to separate the pool of visionary projects from regular operations and rank them according to their risks and estimated potential.

4.5.7 Decision making as an incremental process

In the previous sections, the most relevant perspectives on product roadmap decisions have been discussed along with methods or models to address these. Ultimately, the models provide a multitude of decision criteria expressed either qualitatively or quantitatively (e.g. financial value of a project), with different scale types (nominal, ordinal, interval, ratio). How can these decision parameters be aggregated to make a final decision? We propose a method that allows for both informal and fully structured decision making. Structure and details are added only where priorities are not clear and mutually agreeable. We will first present the fully structured and formal approach before discussing which aspects might be omitted in favor of informal and fast decision structures.

4.5.7.1 Structured decision analysis process

Methods for multi-criteria decision analysis (MCDA) have been proposed to make complex decisions in situations with numerous, potentially even conflicting evaluations. Some of the most popular methods are "Analytical Hierarchy Process" (AHP) [74][75][76], "Preference Ranking Organization Method for Enrichment Evaluation" (PROMETHEE) [14] and "Potentially all pairwise rankings of all possible alternatives" (PAPRIKA) [40]. Unlike other methods, MCDA does not require measurements on all decision parameters but rather focuses on relative strength and preference.

We propose to apply concepts of AHP to the roadmap decision problem. AHP has been widely used in the past decades, and its results, benefits and pitfalls are well documented. Furthermore, several decision making software systems exist which assist in implementing the method in practice. The core idea of AHP is to divide criteria hierarchically and to perform pair-wise comparisons of both the criteria and the alternative solutions with respect to these criteria. The overall results of the pair-wise comparisons are then aggregated. For the pair-wise comparisons, the basic scale listed in Table 7 is applied.

Intensity of Importance	Definition	Explanation
1	Equal importance	Two elements contribute equally to the objective.
3	Moderate importance	Experience and judgment moderately favor one element over the other.
5	Strong importance	Experience and judgment strongly favor one element over the other.
7	Very strong importance	One element is favored very strongly over another; its dominance is demonstrated in practice.
9	Extreme importance	The evidence favoring one element over another is of the highest possible order of affirmation.
Intensities of 2, 4, 6, and 8 can be used to express intermediate values. Intensities of 1.1, 1.2, 1.3, etc. can be used for elements that are very close in importance.		

Table 7 - AHP Fundamental scale for pairwise comparison

Source: Adopted from Saaty: "Fundamentals of Decision Making and Priority Theory" [74] .

The framework can be applied in the light of the above-discussed decision perspectives by seeing each of these perspectives as a high-level decision criterion and the relevant aspects to it as sub-criteria. Based on these, the ranking process is performed. Table 8 summarizes the most relevant decision parameters from the previous sections.

	Criterion	Relevant questions	Applicable models
Financial perspective	NPV	What is the expected ROI?	DCF analysis
	Real options value	What is the value of quantifiable real options associated with a project?	Real-options analysis (Black-Scholes formula, see section 4.5.1.1)
	Virtual options	Are there identifiable but not quantifiable options for expanding, recombining or reusing the product or parts of it? Can the technology be used for other market applications than the current use?	*Use key questions for orientation (see section 4.5.1.1)* ISAA method [49]
	On-costs	What is the project's impact on post-launch maintenance effort (in a period X)? Does it increase / decrease expected maintenance overall? How strongly will maintenance increase with more licenses being sold?	Maintenance efforts reference model (see section 4.5.1.2) Individual estimation of maintenance efforts
	Target costs	How attractive is the opportunity in terms of target costs and estimated returns? Which potential excitement attributes can be implemented within the scope of targeted costs? What is ratio of basic attributes needed to performance and excitement attributes within the target scope?	Target costing and scoping in association with Kano model (see section 4.5.1.3)
Market perspective	Customer value creation	For which customers does a project generate value? Does a product fulfill all basic customer needs? Which basic / performance / excitement attributes does the project contribute? Will the project contribute to the attraction of new customer? Is a project target towards known customer needs?	Kano model [47] (see section 4.5.2.1) Conjoint analysis [35] User-oriented design
	Product maturity	How much will an enhancement contribute to existing customers' satisfaction? How mature is the product already in terms of requests, reported bugs and feature scope? How is the product perceived in the market place?	S-curve theory (see section 4.5.2.2)
	Competitive position	What is the contribution of a project relative to competitors? Is it required to keep up with or to outperform competitors? After project implementation, will there still be better offerings in the marketplace? What is the relative strength of a product / product line relative to competitors?	QFD [1]
	Market impact	What market is addressed with a project? Do we have access to the market? What is the probability to generate visible impact in this market? When will the project show results in the market? Which are uncertainty factors and potential risks for a project?	-

Stakeholders perspective	Channel fit	Does the product meet the needs and fit the requirements of all stakeholders along the sales and distribution channel? Can the project's value easily be communicated?	*Use key questions for orientation (see section 4.5.3)*
	Appeal for stake-holders	How attractive is the project from the perspective of other stakeholders? Is it attractive to build, feasible to maintain and service etc.?	
Technology perspective	Technology platform	On which platform / base technologies does a project build? How sustainable are these, and does the project conform to recent developments in the field?	-
	Technology trends	Does the project fit current high-level technology trends and hype cycles in the marketplace?	
	Technology leverage	Does the project leverage external sources?	
Operations perspective	Resources and constraints	Which resources are required and are these available? Which impact does the project have on other projects and the overall resource utilization?	Roadmap planning and scheduling (see section 4.6.3)

Table 8 - Decision parameters and related models to gather input data

The result is a hierarchy of criteria for the decision process, as schematically illustrated in Figure 12. As the structure grows rapidly with the number of decision parameters being considered, we propose selecting them carefully, i.e., starting at the high-level goals, and including sub-criteria on demand if the raking between individual projects according to a criteria is unclear. For instance, Figure 12 indicates the use of four high-level criteria, whereas sub-criteria such as used for analyzing the financial impact in detail are used on-demand. Their results are aggregated to the overall rating for financial impact.

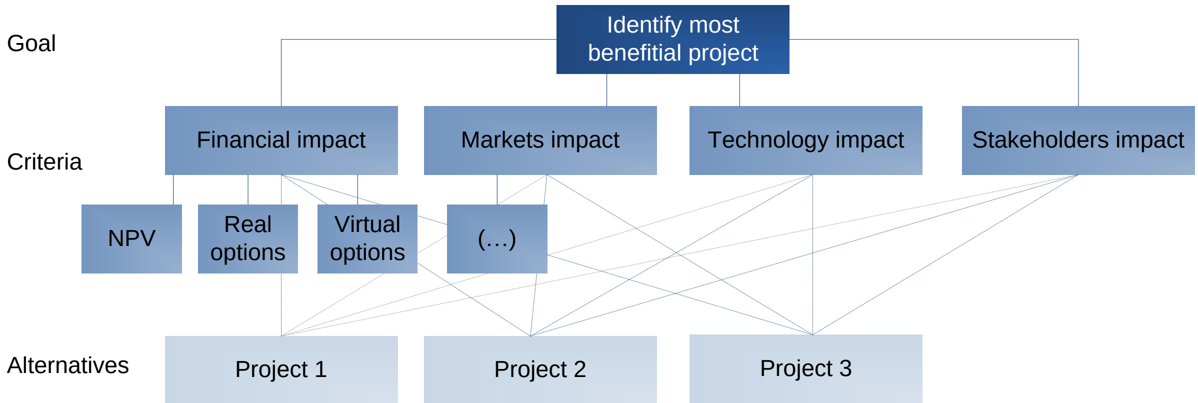


Figure 12 – Criteria hierarchy for roadmap decision making

According to the AHP, the decision process is carried out by performing pair-wise comparisons of the alternatives according to each criterion. Table 9 shows an example based on a template proposed by authors of the method.

Project 1	1	Project 2	4	Reasoning: (…)
Project 1	4	Project 3	1	…
Project 2	9	Project 3	1	…

Table 9 - Pair-wise comparison with AHP method [74]

The priorities are aggregated using a comparison matrix (see Table 10). Mathematically, the matrix' principal right Eigenvector is computed. Furthermore, Saaty proposed to compute an inconsistency indicator to show whether the pair-wise rankings contain transitive inconsistencies. Further details on the method can be obtained from Saaty's basic texts on AHP [74][75][76].

	Project 1	**Project 2**	**Project 3**	**Priority**
Project 1	1	¼	4	0.217
Project 2	4	1	9	0.717
Project 3	¼	1/9	1	0.066
			Sum of priorities	**1.000**
			Inconsistencies	**0.035**

Table 10 – Example for aggregation matrix for pair-wise comparisons in AHP

Source: Structure and method adopted from examples provided by Saaty [74][75]

This step is repeated for every criterion. Then, the criteria by themselves are evaluated against the goal in the same way. The result of these steps will be (1) a numeric ranking of the criteria, and (2) a numeric ranking of all alternatives according to each of these criteria, which can be combined computing the weighted product.
Ultimately, the result is a fully ranked list of all alternatives on a nominal scale, i.e., with a measure of "distance" between the alternatives (see Figure 13).

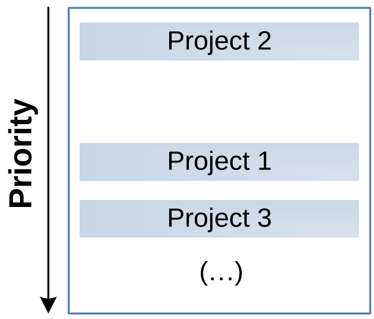


Figure 13- Result of basic AHP method: Ranked projects list

4.5.7.2 Informal and semi-structured decision making approaches

Despite the benefits of a structured decision analysis method, the process might become lengthy when dealing with large decision trees and a large number of pair-wise comparisons. We therefore propose more light-weight alternative methods, which consider above-said decision perspectives without doing deep-dive analyses. Depending on the effort one is willing to spend for the decision process and the targeted accuracy and reliability of results, a method in the spectrum from an informal ranking process, over a ranking per decision criterion to the full AHP process with pair-wise comparisons can be applied (see Figure 14).

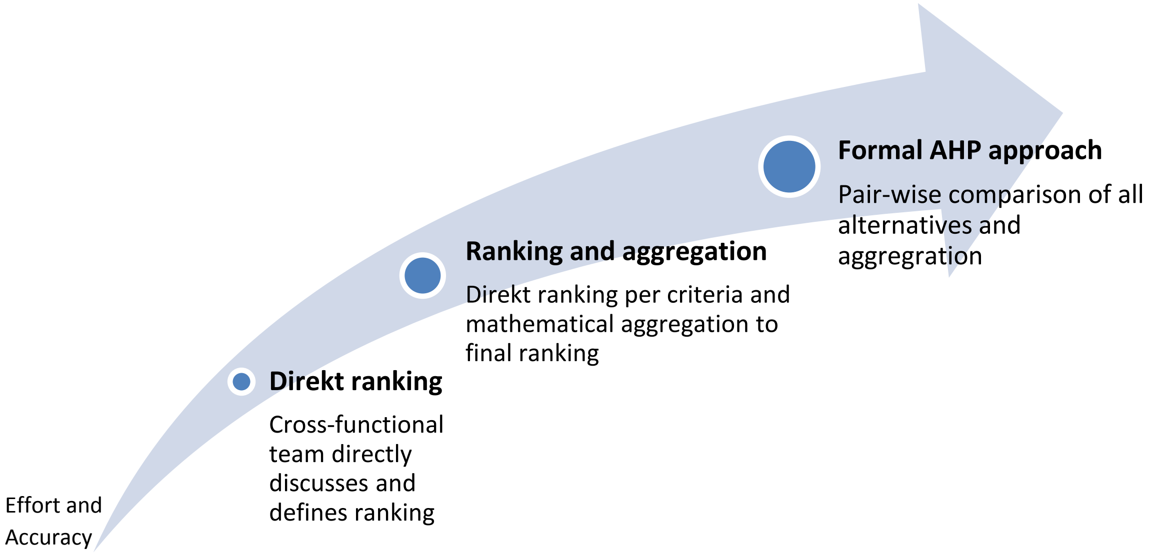


Figure 14 - Spectrum from informal to structured decision making

In the informal, direct ranking process a cross-functional team is formed as a roadmap committee which regularly decides over roadmap updates. We propose product management to take the lead role in this committee. Attendees should represent product design, development, product support, marketing, and sales (and other groups if relevant). In regular meetings, the ranking of projects is determined and continuously updated if new projects are to be added or new information emerges. The stakeholders represent their perspective on the decision process. Product management proposes an initial ranking of projects. Subsequently, projects are re-ranked based on consensus decisions.

In the semi-structured approach of ranking and aggregation, the first step of pair-wise comparisons in the AHP is skipped and instead, the informal ranking process is applied. In contrast to a fully informal process, the ranking is done with respect to each relevant decision criterion based on the key analysis questions, and then the overall ranking is aggregated based on the weights of each criterion. The semi-structured approach can also be used in combination with AHP: for particular criteria where priorities are mostly clear the ranking might be established directly. Only if the ranking for a criterion is unclear or not mutually accepted the process of pair-wise comparisons is applied to establish the ranking.

4.5.7.3 Multiple ranking pools

We stated earlier that in practice there is a need to manage a constant flow of activities stemming from ideas or projects in different stages of their lifecycle. Consequently, projects being in different states of their lifecycle cannot always be aggregated into one ranking. Rather, a separate ranked list is maintained as soon as the ranked items are either incomparable in nature (e.g., an open research project vs. an implementation project) or require distinct resources and thus do not interfere. Our model foresees the definition of n ranking pools either based on resource pools or on lifecycle stage or a combination thereof. An example is proposed in Figure 15, where three ranking pools are defined: opportunity assessment activities are carried our according to a ranking of ideas, functional and technical analysis, design and prototyping activities are carried our according to a ranking of concepts, and development tasks are done based on a projects ranking.

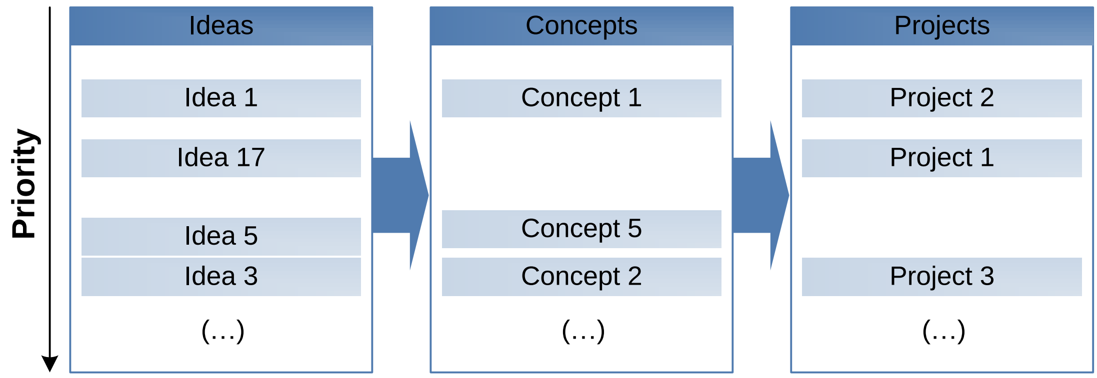


Figure 15 – Proposal for ranking pools along product lifecycle

4.5.7.4 Integration into product lifecycle management

Figure 16 below illustrates how the ranking process integrates into continuous product lifecycle management. The decision perspectives provides key decision criteria and models which allow to perform the ranking process per criteria either based on quantitative figures or a qualitative pair-wise ranking. The overall ranking of projects is established by applying the structured AHP decision analysis method or the informal/semi-structured approach discussed above. Based on the priority ranking and the resource needs of each project, an ideal schedule for operations can be derived (details on this process are discussed below in section 4.6). Innovation controlling mechanisms and metrics provide real-time visibility on the state of the process and its results. These metrics are used to continuously refine the priorities in the AHP, i.e., the list and ranking of relevant criteria. If the structured decision method is not applied, controlling metrics provide valuable input for the roadmap committee. Operations provide novel information, updates or project parameters (e.g. status updates from implementation, effort estimates, technology evaluation results), which are used to readjust the AHP model.

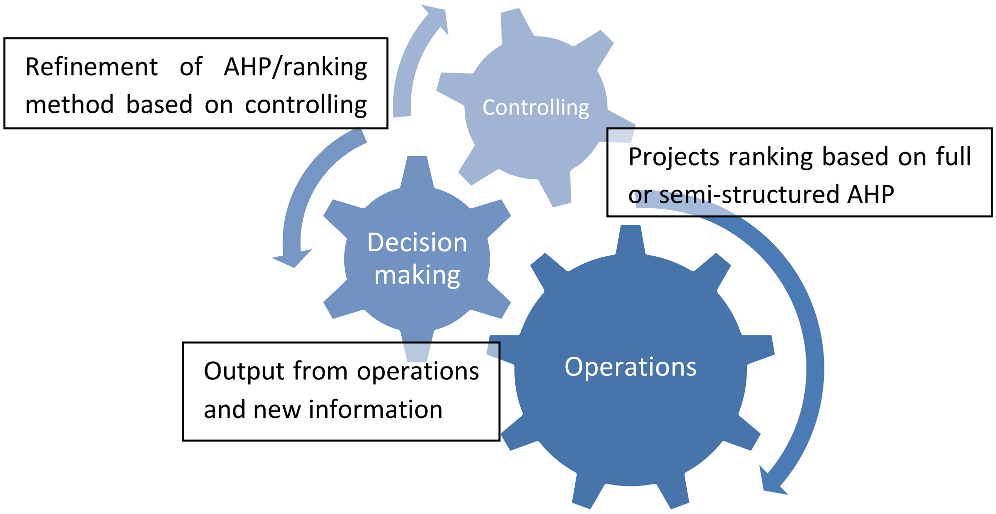


Figure 16 - Integration of decisions, operations, and controlling

4.5.8 Balancing the product portfolio

The techniques presented thus far support the assessment of relative project benefit and as stated earlier the result is a ranking of projects by their value. In practice, however, simply investing in the top priority items by their ranking might not be a viable option. For instance, the top-value projects might concentrate the risks strongly in one direction, or do not allow to show regular outputs and product releases. Goffin et al. [34] summarized several techniques for balancing the R&D portfolio, which we adopt and extend.

4.5.8.1 Balancing timing and the delivery flow

First, Goffin et al. [34] propose to balance the timing of outputs from R&D projects. Generally, it is desirable to keep a continuous stream of outputs. When planning the product roadmap, it should be defined beforehand, when which outputs will be generated, what their USPs and impact will be, and what will be actively marketed. There might be cases where concentrated output is desired, for instance for important trade shows, but generally, a continuous and regular flow best supports launch activities and balances resource needs for launch, marketing, consulting, training and maintenance.

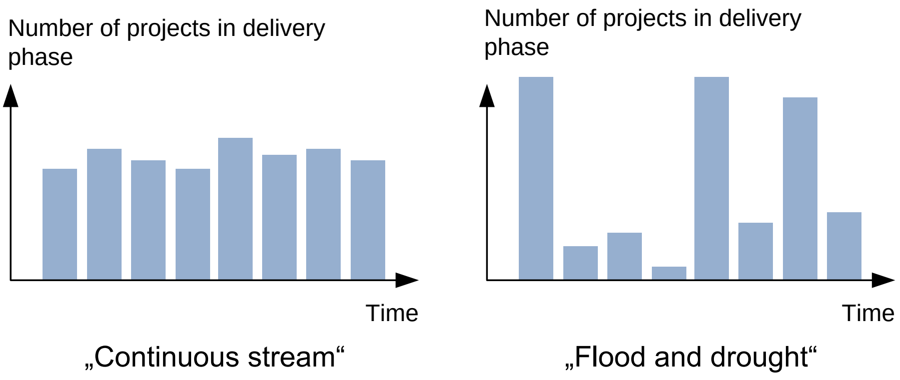


Figure 17 - Output timing in project portfolios

Source: Goffin et al., 2009 [34].

4.5.8.2 Balancing risks and the features mix

A risk/reward portfolio (see Figure 18a) allows to keep a balance between high-risk/high-return and low-risk/low-return projects, allows to identify the pearls of low-risk/high-return projects and to avoid high-risk/low-return operations. Projects in the high-risk/low-return quadrant should have the lowest ranking based on the decision process. Furthermore, balancing the portfolio of projects in terms of project scope (see Figure 18b) helps in keeping a mix of minor and major enhancements and eases the establishment of a continuous delivery stream as discussed above. Adding to these, we identified the necessity to balance the investments in individual product features versus the investments in product platforms (see Figure 18c). Individual product features typically rank much higher in terms of their financial valuation, market impact and general attractiveness and relevance for stakeholders, but keeping the product platform updated is vital for future enhancements. Cagan [17] refers to these investments as “technology heads-up” and argues to keep the base platform in the lead as compared to its usage by individual features build on top.

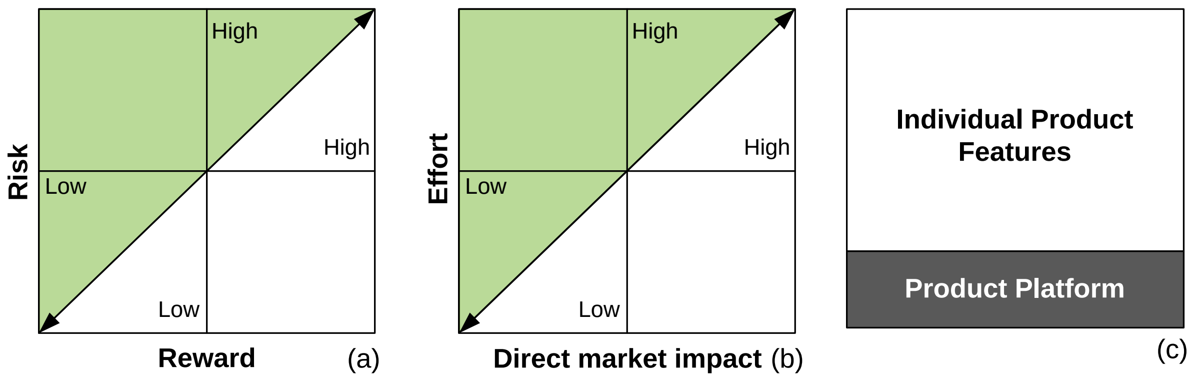


Figure 18 - Relevant matrices to balance the project portfolio

4.5.8.3 Balancing the strategic resource allocation

In addition to defining and refining decision criteria, we propose the definition of a target resource allocation for the different ranking pools, i.e., which quantity of the total R&D resources will be used for activities in each pool. Often, firms already have such an allocation defined and it is frequently found in executive reports from the R&D department: How are R&D resources spend. Defining this allocation in advance and monitoring it continuously is an important step towards an optimal resource allocation. An example for a target resource allocation from a software firm is shown in Figure 19.

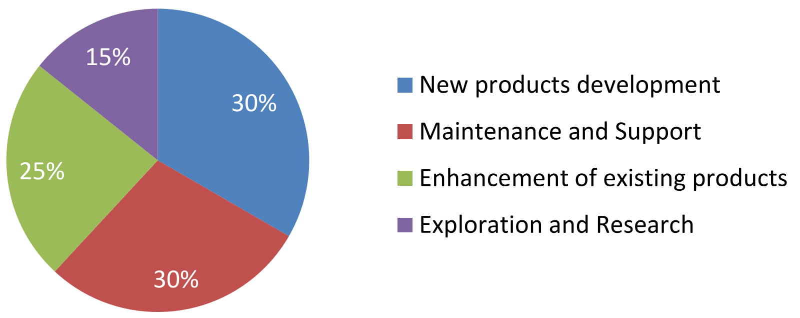


Figure 19- Example for a target resource allocation in a software firm

The target resource allocation is also part of the decision process and treated similarly to the decision criteria. The allocation percentages implicitly define a ranking for resource investments, which is also continuously evaluated and adjusted if necessary. Goffin et al. [34] documented a similar concept of so-called strategic buckets being used at the insurance company AXA Insurances Ireland applied for both orientations of projects by markets and by general characteristics.

In particular, the strategic bucket for exploration and experimentation is relevant in this context. We argued above, in section 4.5.6 on strategic foresight, that there is a tendency to neglect long-term changes and cut resources in favor of urgent, short-term activities. The strategic bucket and controlling of the resource being invested ensures that the long-term focus is being accounted for.

In practice: How to define the strategic resource allocation?

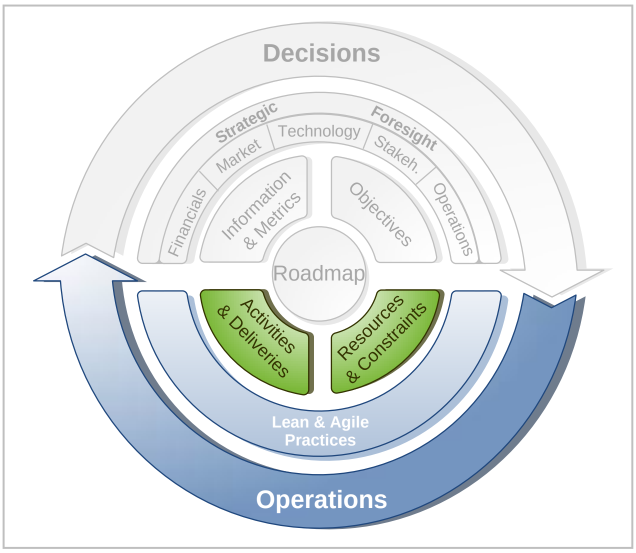
In practice, the strategic resource allocation has significant impact on a firm's success on the long run. Thus, defining or adjusting it is a highly critical issue. We recommend starting with the current allocation of resources and identifying a realistic rate of change for making adjustments. How flexible are resources? Which resources are bound to certain buckets and cannot be shifted? Starting from there, flexible resources can be dedicated towards new products, exploration and research based a firm's willingness to take risks, current market position, sense of urgency for change and other factors.

We further recommend defining the strategic resource buckets first, before deciding over the roadmap projects. This guarantees that the buckets are defined independently from project candidates and not adjusted to fit-in certain projects.

4.5.9 Management power and politics in roadmap decisions

Throughout the past sections, we presented a structured and objective approach towards product roadmap decision making. Still, we do not want to neglect the fact that in practice, roadmap decisions are not only driven by qualitative and quantitative analyses, numbers and facts but are often subject to corporate politics, power games, management rivalry, or individual interests resulting from compensation schemes, complex linkages and networks within organizations and the like. These aspects are out of scope for this thesis, but the interested reader may refer to Weissenberger-Eibl and Teufel [97] for a more detailed discussion on these issues.

4.6 Operations



The decision and prioritization model discussed in the previous section continuously determines the most beneficial projects and ideas to be pursued. Operations need to be aligned with these priorities and prepared for a continuous (re)validation process and subsequent changes. In the model overview, we described an alternative view on NPD (see section 4.1), as a continuous stream of activities for projects being in different stages of their lifecycle from the fuzzy front-end of innovations to post-launch maintenance and support. Depending on the lifecycle stage, different success factors drive operations (see Figure 20). Intuitively, the nature of the activities and the involved stakeholders vary along the lifecycle stages. This fact makes lifecycle management particularly challenging from an operational perspective. The high number of stakeholders results in a high number of interfaces and needs for synchronization in order to achieve short time-to-success cycles. Bottlenecks and resource overloads may not only impact one, but several projects sharing certain resources.

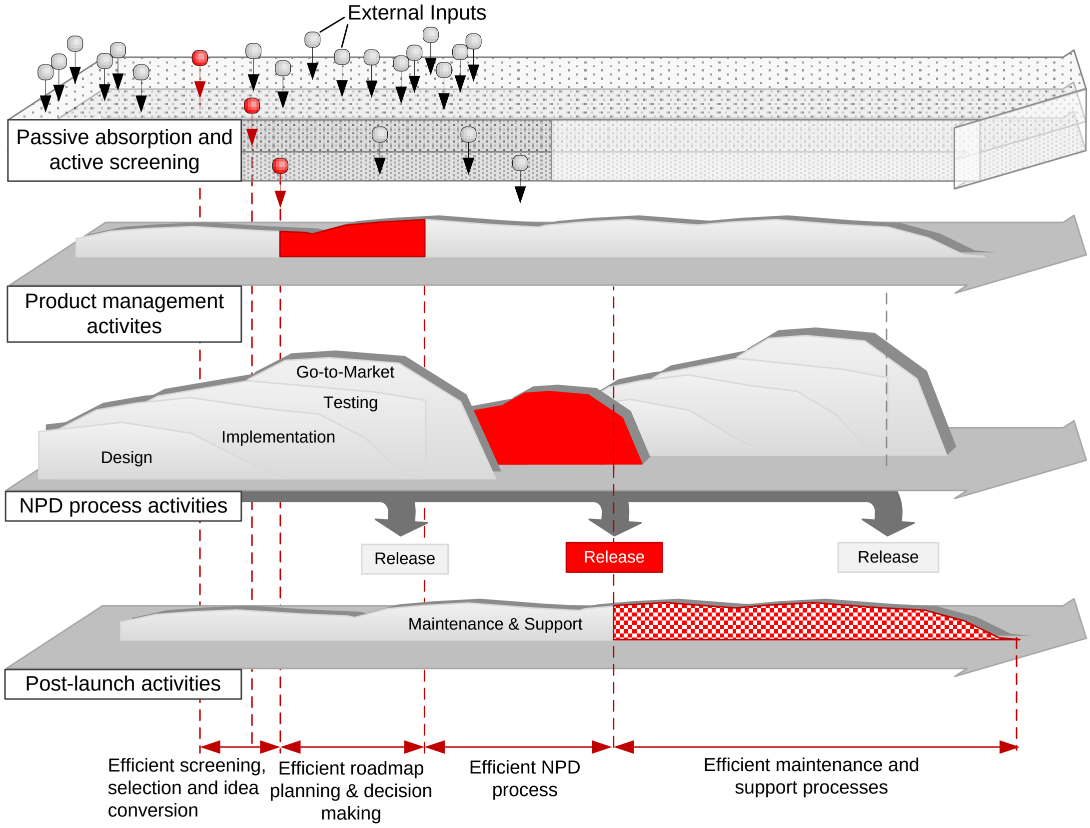


Figure 20 - Key activities and success factors along the product lifecycle

Another major concern in operational roadmap planning is how to achieve transparency over resource implications stemming from roadmap decision, particularly when it comes to short-term changes. Often, highly urgent tasks and features are added, but the consequences for the roadmap (i.e., to which extent does it change the overall schedule and delivery) are not clear. One major reason for that is thinking of resources in terms of a global pool that can be flexibly allocated to any activity, whereas in reality, resources are compatible only with certain tasks, complex dependencies exist in their tasks, and running into a bottleneck at one stage delays the whole project. The proposed roadmap scheduling model reflects these resource constraints transparently and shows the consequences of adding or removing items to the roadmap, or reacting to emergent information.

4.6.1 Operations model overview

Based on principles of lean production, we propose a model for NPD operations aligned with the presented decision framework. The model is based on two core components. First, activities and priorities are managed via a ranked backlog of tasks for every lifecycle stage. These priority queues are continuously updated with every product roadmap decision being taken. Second, resource availability and constraints are modeled and resource compatibility is determined, i.e., which resources can be used in which projects. Based on this information, the schedule of tasks for every resource is derived.

Figure 21 illustrates the process conceptually from an unstructured pool of ideas to a product shipment. The number of backlog queues is not fixed and can be adjusted either based on involved stakeholders (for instance, one queue per team may be used), projects or the strategic buckets for resource commitments, which allows then to dedicate a certain number of resources to a queue.

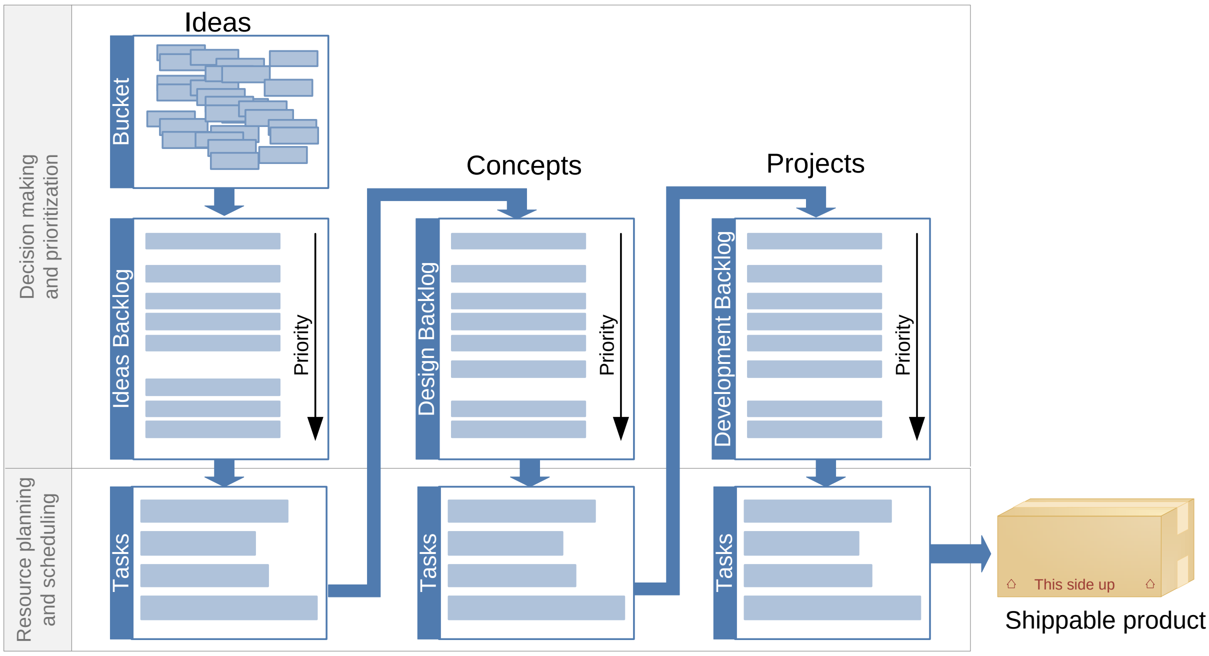


Figure 21 - Flow of activities through product lifecycle

4.6.2 Managing work queues

For managing the individual work queues, we refer to best practices and the body of knowledge available on lean production and lean software development. Poppendieck and Poppendieck [69] discuss relevant key concepts such as elimination of waste, late decisions, amplification of learning, and the concept of local responsibilities vs. global optimization.

We generally see an efficient pull mechanism from the task queues and limiting work in progress (WIP) as key success factors. With that, an efficient value stream without idle times and bottlenecks is possible. Furthermore, we see a strong need to invest in resource flexibility. The more flexible and universally assignable a resource pool is, the better can the load be balanced, particularly in times of high uncertainty and little knowledge on domain and technology know-how being required for the next product generations.

4.6.3 Resource planning and task scheduling

Priority queues as used in lean production management guarantee resource investments in tasks with the highest priority at any given point in time. Beyond that, we discussed above that an operational perspective on product roadmap decisions requires to constantly point out resource implications of roadmap decisions and to consider resource availability and constraints in the decision process. An overall schedule, derived from an ideal allocation of tasks across the queues given all resource constraints is required. The scheduling process is designed to maximize the throughput of projects with the highest rank using the available resources. Figure 22 shows an example for a ranked list of projects and features.

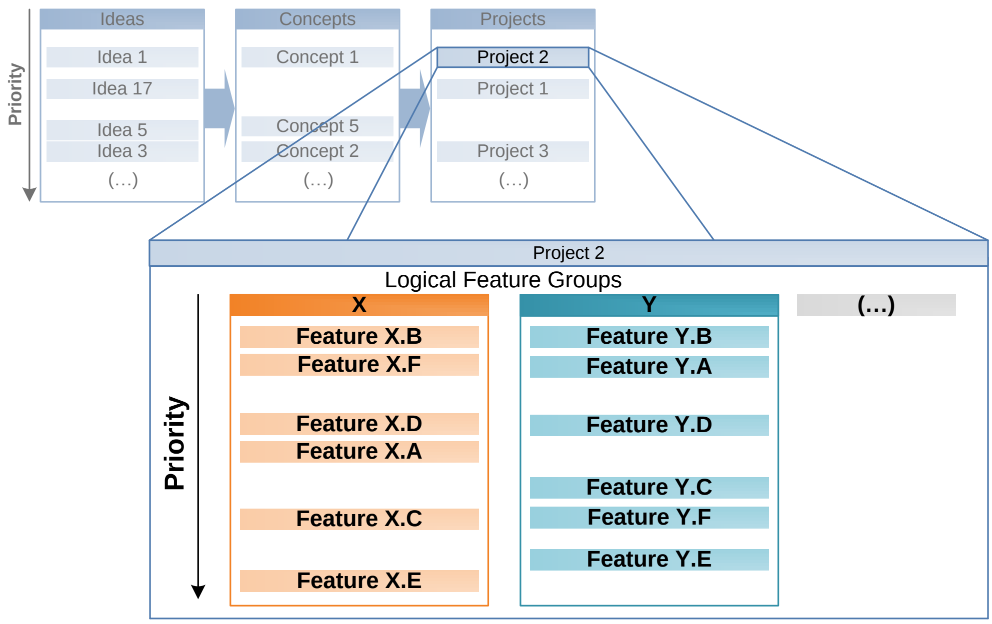


Figure 22 – Ranked features list by logical feature group

The ranking might be a plain order, or allow for an additional distance function in the ranking, in order to indicate major differentiations in priority between subsequent items.
From practical observations in software development, we further propose that within a project, logical sub-groups of features are established based on feature types, or, in case of distinct and not interchangeable resource pools, based on these groups of resources.

The next input required for the roadmap scheduling model is resource availability and compatibility. This is achieved by establishing (1) a resources/skills matrix indicating who is capable of accomplishing certain types of tasks (see example in Figure 23), (2) a resource schedule indicating resource availability over time, and (3) a resource occupation estimate, indicating which projects take which quantity of resources (see Figure 24).

In practice: Managing resources and skills

In practice, skills matrices often exist within firms, but are rarely up-to-date. The idea of knowing exactly which skills exist where within a firm, and to access them on-demand is not new. However, the implementation often fails due to the scope and complexity of these matrices, making frequent updates tedious and unfeasible. We recommend keeping the matrix and the number of distinct skills being tracked very narrow and focused on really distinguishing items such as programming languages or design skills. Keeping track of who knows which version of an operating system or other fine-grained items will rarely be required for the planning process.
Generally, establishing resource elasticity becomes a core objective in order to cater to short-term needs flexibly. Yet, there should be stability in resource planning for long-term, sustained engineering efforts.

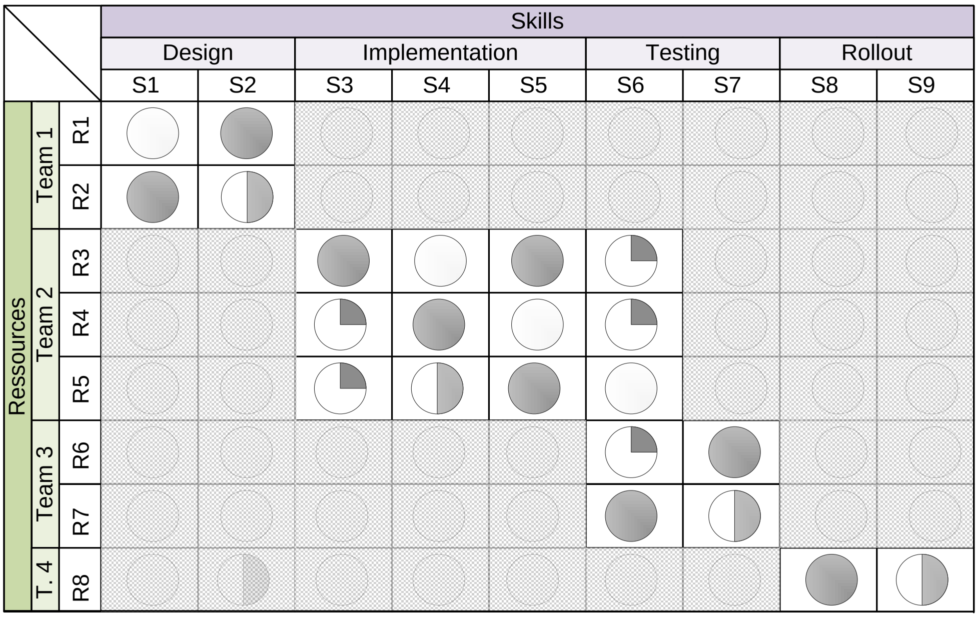


Figure 23 – Resources/skills matrix for different functions

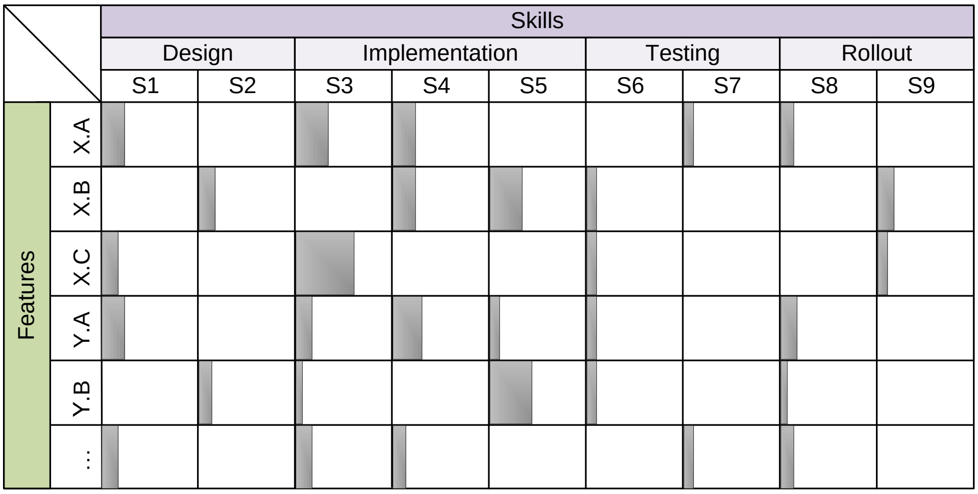


Figure 24 – Schematic illustration of required resource levels for all projects

The required resource levels are combined with actual effort estimates for every project and an indication for the expected variance in the actual effort, i.e., deviation from planned vs. actual efforts, or a direct estimate for best, worst and average cases.

Feature	Effort best case	Effort worst case	Effort avg. case
X.A	30	45	35
X.B	40	60	50
....			

Table 11 – Example for effort estimates in days for selected features

Finally, the minimum and maximum number of parallel projects has to be defined, depending on the size of the involved teams and project character. Yu et al. [102] have studied the relationship between the complexity of a product family and the number of product families that can be launched per unit of revenue and observed a significant correlation, i.e., the more complex a firm's products the less parallel projects are possible. In practice, we observed a "natural boundary" for parallel projects which is known from experience within each team in the NPD process and may vary strongly from firm to firm and team to team. For instance, within a software development group the upper bound p_{max} was roughly estimated as:

$$p_{max} = \frac{\#team\ members}{2}$$

I.e., at least two team members are assigned to a project on average. The lower bound p_{min} was estimated to be:

$$p_{min} = \frac{\#team\ members}{5}$$

I.e., maximum five people work together on one item, beyond that it is split into distinct enhancement projects.

Based on these input parameters and additional, manually configured preferences and constraints an optimized schedule can be derived or computed. We propose applying an optimization algorithm for computing a proposed schedule on-the-fly. The requirements can be mapped to the well-documented and researched *flexible job-shop problem*. Numerous algorithms have been proposed for computing an optimal schedule based on given resource constraints, including simulation-based approaches [101][103], approaches based on swarm intelligence [56] as well as local search and taboo search [45][55] and genetic algorithms [95].
The optimization algorithm can be configured towards optimizing for maximum throughput of high-priority projects or maximum output at a certain point in time. When configuring to **finish top n projects as soon as possible** the throughput on high priority projects is strictly controlled. Other projects will be started only if resources incompatible with the required skills of the high-priority projects are still available, or the minimum number of parallel projects is not reached. When configuring to **maximize the number of projects finished by date x**, the algorithm will maximize the presentable output, thereby potentially postponing higher priority projects if they block too many resources. This mode is applied if it is necessary to demonstrate fast and broad market impact rather than a few high-profile projects, or to achieve the above discussed regular outflow of deliveries.
Figure 25 shows an example for a schedule derived based on defined resource pools, constraints and priorities.

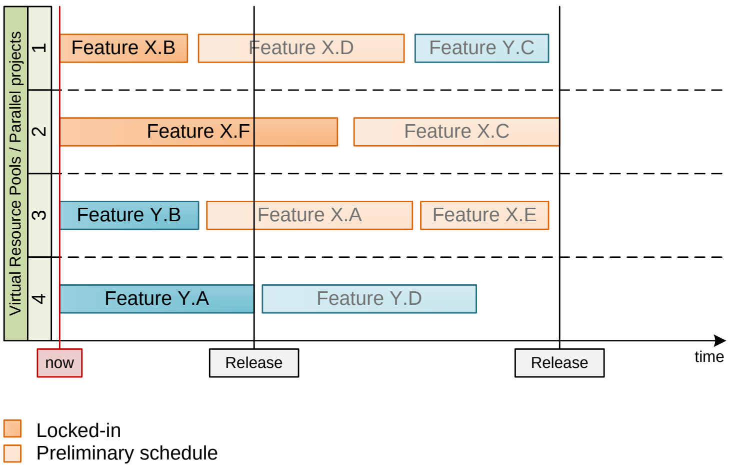


Figure 25 – Optimized project schedule derived from resource model and priorities

4.6.4 Dynamic rescheduling and roadmap I/O

The computed preliminary schedule is the basis for what we refer to as the *roadmap I/O*, the transparent "in-and-out-mechanism" for roadmap changes. In the light of high uncertainty, a flexible production environment, and frequent decision making, we consider this mechanism as key to keeping decision making and operations aligned. The roadmap I/O mechanism guarantees that resources are invested in pursuing the most valuable projects at any point in time, without interrupting the stream of regular deliveries, and at the same time avoiding sunk costs stemming from abandoned projects. The I/O model allows to flexibly add projects to the roadmap and guarantees that the least preferred project will be postponed accordingly, in order to avoid resource overload situations. In addition, the scheduling mechanism supports the analysis of what-if scenarios to assist the roadmap decision and prioritization process. Updates to the schedule are done in any of the following events: (1) a new project is added to a queue, (2) new information on a project's benefit and value emerged and a re-ranking is done, (3) remaining effort estimates are updated, (4) emergent requirements or new information leads to scope changes, or (5) resource availability changes. Figure 26 shows an example for dynamic rescheduling in case of a new project being added to the top of the features list. In order to avoid sunk costs, features currently in progress are locked-in, and will thus not be rescheduled (except in special cases, if ultimately necessary due to market constraints, a limited window of opportunity, or resource changes etc.). All other features are scheduled only preliminarily, and the new feature will be inserted into the most suitable resource line according to the optimization function defined.

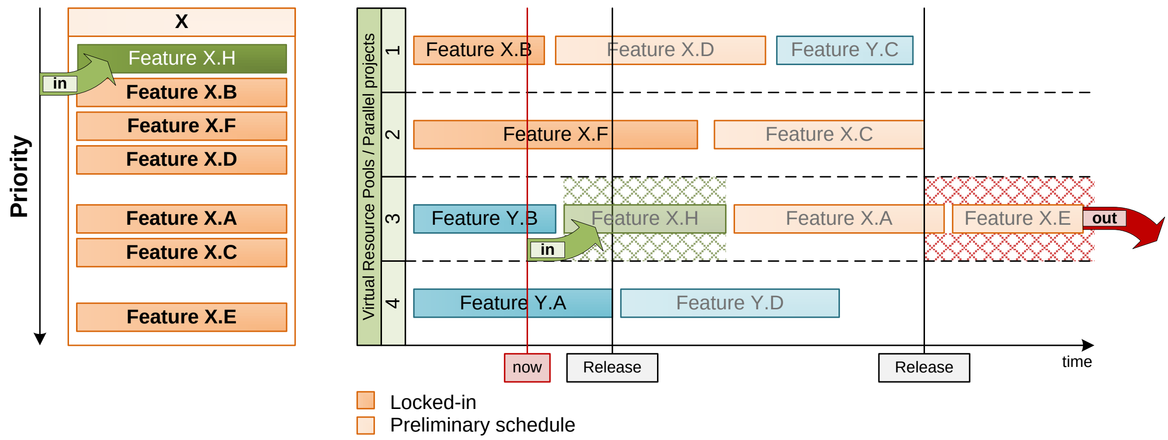


Figure 26 – Dynamic rescheduling in case of priority changes or emergent information

The main benefit about this simple process is its transparency for all stakeholders. It is immediately visible which features drop out of the release scope when adding new items or adjusting to new information being available.

4.6.5 Release planning with variable release pace

The roadmap scheduling mechanism is the basis for defining release dates and preliminary scopes of these software releases. In many software firms, there are fixed release policies for major releases, minor releases, and service packs. These policies define the release schedule, as well as which release types might contain new features, or which contain only bug fixes. For instance, a firm might deliver major releases every two years, containing large new features and significant changes to the system, minor releases every six month, containing only small enhancements, and service packs monthly to fix critical bugs. This policy has to be defined based on customers' capacity to absorb changes and perform updates. The roadmap schedule has to consider these release cycles.

In order to still deliver components frequently to the marketplace, we recommend planning at variable release pace for different product components. Modular design enables this flexible pace in order to release frequently on peripheral system components and less frequently on core components and the platform. Modularity is thus the enabler for combining sustained engineering with fast innovation cycles.

4.6.6 Operational constraints

The basic scheduling model described above assumes that all projects and features are independent from each other and in no temporal order, and that their business value remains unchanged, whether the project is scheduled now or after a certain time period. In practice, however, features might be dependent on each other, and there might be other limitations such as a window of opportunity in which a development project has to be accomplished in order to realize the associated benefits. Due to these considerations, the following extensions to the basic scheduling approach are proposed.

4.6.6.1 Feature dependencies

The model allows defining feature dependencies in terms of logical predecessors/successors. Figure 27 illustrates the types of relations possible, including sequences of $2..n$ features (a), $1 \ldots n$ predecessors (b) and overlaps (c), as well as a combination thereof.

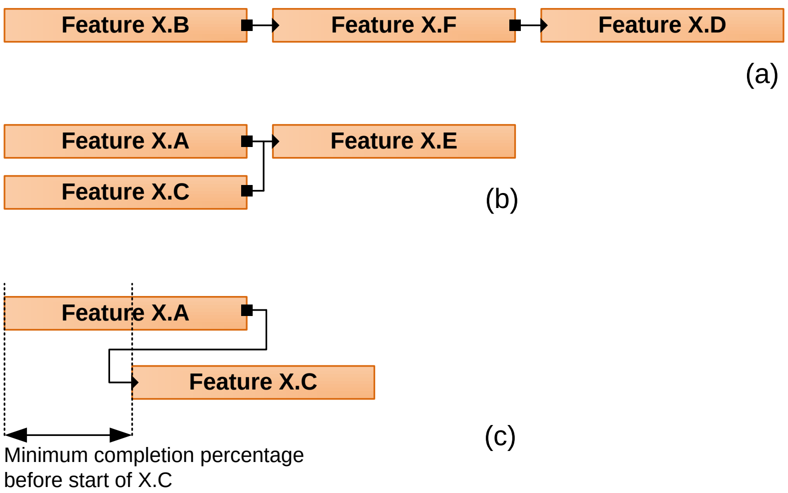


Figure 27 – Options for modeling dependencies between features

In practice, the number of these complex dependencies shall be kept at an absolute minimum. Roadmap projects should be designed and treated at a granular level in order to allow for an independent judgment of priorities. Yet, the more fine-grained the roadmap scheduling becomes, the more dependencies will exist.

4.6.6.2 Subtasks and task phases

> ***In Practice: Planning to fail early***
>
> *In practice, sub-tasks should not only be used to do more fine-grained planning and to schedule more activities in parallel. We see a strong need to use these mechanisms to work in a risks-first approach, i.e., to schedule the most uncertain activities first, to perform actions that reduce uncertainty or reveal novel information, and thus in case of wrong assumptions to fail as early as possible.*

The skills matrix illustrated above shows skills for different stages along the NPD process. Typically, these resources are not overlapping, and may originate from different departments of a firm, such as design, development, quality assurance or marketing. When planning only in overall feature pools, we observed weaknesses in terms of potential bottlenecks at tasks concentrated strongly at the beginning or the end of a project. The reason is that the model is designed to start several projects in parallel from a certain point in time, and have them ending at the same point in time, in order to package a new product release. To avoid or at least consider these potential bottlenecks in the scheduling process, the explicit modeling of sub-tasks is required, which can, again, be scheduled in parallel, in sequence, or with a defined overlap.

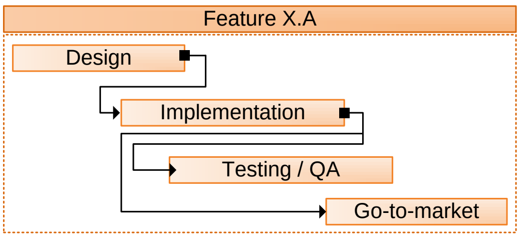


Figure 28 - Overlapping subtasks

Based on the concept of sub-tasks, the model can be scaled up and broken down to any granularity needed, from a high-level roadmap down to the individual tasks of each involved person. An additional challenge for the scheduling process is the unequal resource utilization over time observed in practice. The total time estimated for a project is typically not consumed regularly throughout the complete lifecycle. Instead, often an inception phase can be observed where fewer resources can work in parallel on design and prototyping, before more resources can work in parallel during the implementation and testing phases (see Figure 29a).

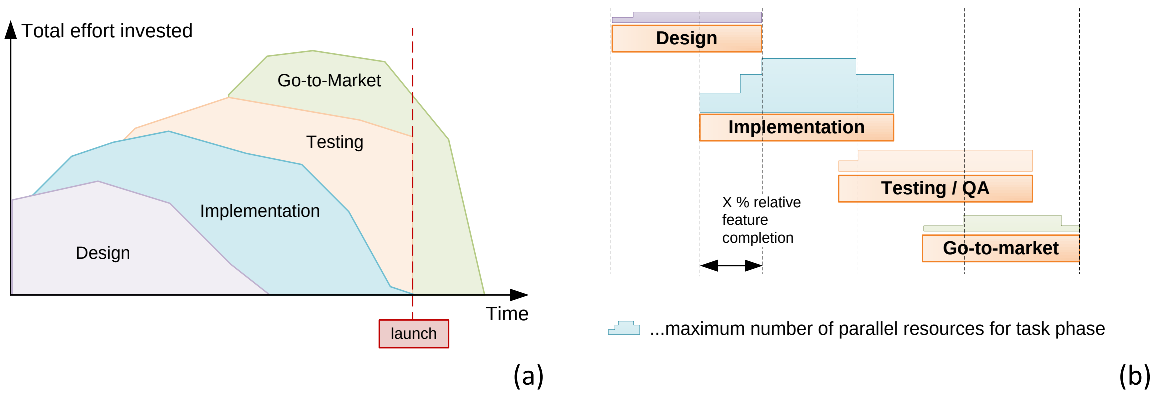


Figure 29 - Invested effort over project lifecycle and phase

In terms of an optimization algorithm, this can either be controlled by modeling the maximum possible number of parallel resources assigned to a certain task over time (see Figure 29b), or by approximating such a load model from historic reference data.

4.6.6.3 Windows of opportunity and time constraints

Certain projects and features only generate value in a defined time period, e. g. due to competitive pressure/catch-up or other factors. Therefore, these projects might have constraints in terms of an earliest start, a latest start, and a latest end date. Another source of time constraints might be deadlines committed to customers, partners and other stakeholders for delivering certain components, or strategic deadlines originating from high-level corporate plans.

4.6.6.4 Resource costs and resource limits

Certain resources might be more costly than others (e.g. an external partner firm implementing a piece rather than an internal developer), but also more flexible. These can be added on demand, but they add additional costs. The scheduling process is able to consider these costs and take total cost limits into consideration.

4.6.7 Scenario simulation to test for schedule robustness

A perfect schedule in the light of all current information might not be sufficient for an ever evolving business environment. We claim that it can even be dangerous to optimize a schedule to an extent where it maximizes returns on invested resources from a current point of view, as it might be sensitive to any kind of parameter change, or its execution depends on few key resources, where deviations quickly delay the whole roadmap delivery. Instead of only optimizing for time-to-market and project throughput, a balancing factor is required to judge how robust the schedule is, e.g. by simulating the absence of certain key resources, small and major project delays etc. The best schedule provides both an attractive best-case scenario and the least disruptive worst-case scenarios. Scenarios are computed in terms of best-case, worst-case or random deviation models. A random simulation of deviations allows judging how robust the selected schedule is towards a certain extent of deviations.

4.6.8 Schedule extrapolation

Finally, the roadmap schedule is also the basis for roadmap extrapolation as discussed in the considerations on strategic foresight in section 4.5.6. We recommend continuing the roadmap from the highest level of detail for current enhancement projects and features to less-detailed projects to broad roadmap topics and roughly sketch the resource usage and expected outcomes. Even though the preliminary schedule will be subject to refinement and change over time, it is a valuable source for orientation and alignment of all stakeholders.

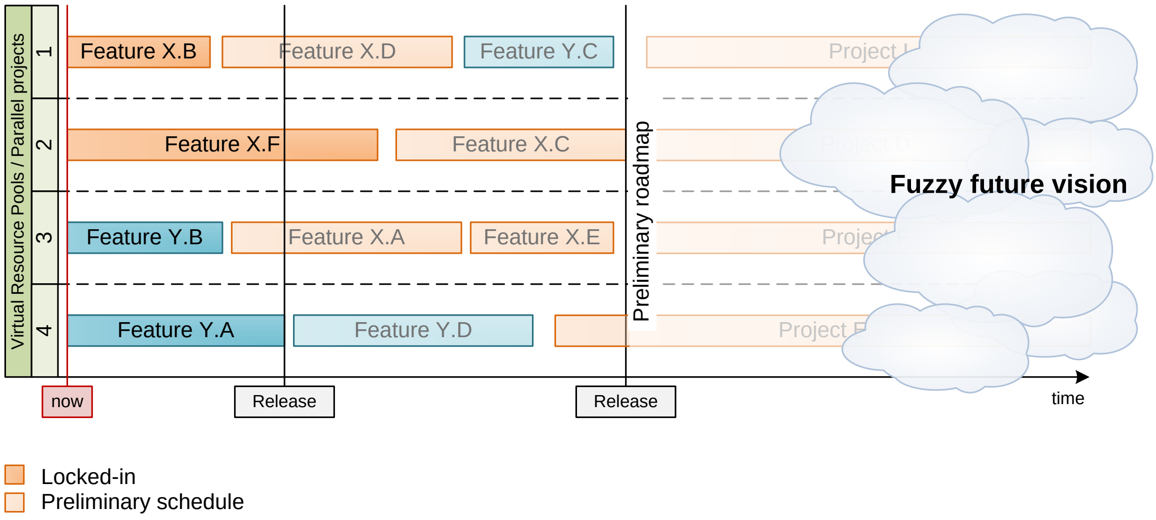


Figure 30 - Roadmap schedule extrapolation

4.6.9 Process model

Based on the proposed approaches towards decisions and operations, we have designed an adjusted NPD process model in order to incorporate the results into daily operations.

4.6.9.1 NPD process

We adopted the core idea of network models (see Trott [90]), suggesting an incremental accumulation and refinement of knowledge, linked with external sources of input. The resulting process model tailored to the specifics of software development is illustrated in Figure 31.

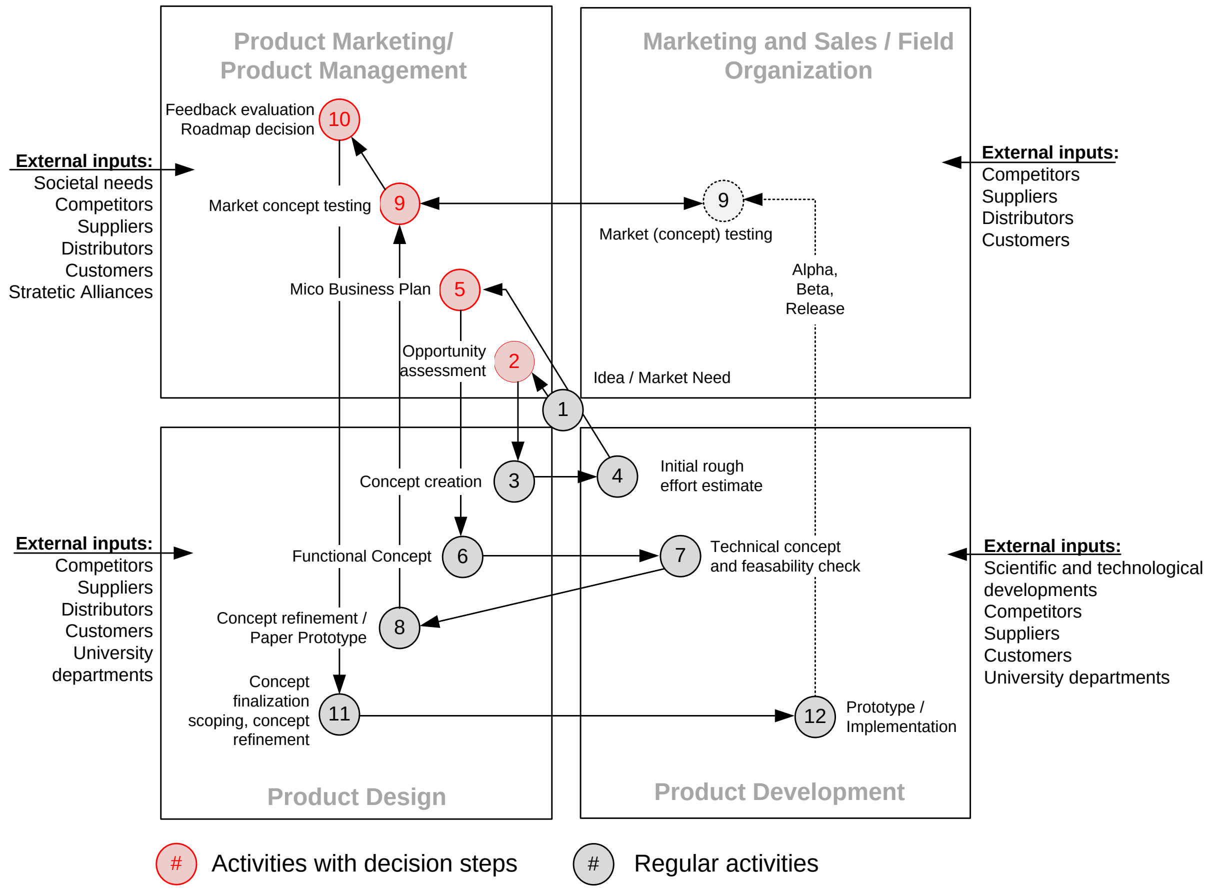


Figure 31 – NPD activities from idea to launch with external inputs to network model

The model does not imply a linear and sequential order of activities. By breaking activities down into individual work packages, these can be done in parallel within the involved groups.

4.6.9.2 Global workflow model

Overall, we see the product lifecycle management as a master process implemented on top of and integrating with integrating with existing sub-processes. The global workflow is illustrated in

Figure 32. Core of the model is the roadmap, maintained by product management. From the roadmap, current work-in-progress (WIP) items are selected based on global priorities and a balanced product portfolio. From these high-level projects, tasks are derived for the individual functions involved in the NPD process such as design, development or quality assurance. Results of these tasks are either intermediary outcomes added to the queue of the next functional group, or intangible results which lead to a reassessment of priorities, or shippable product increments.

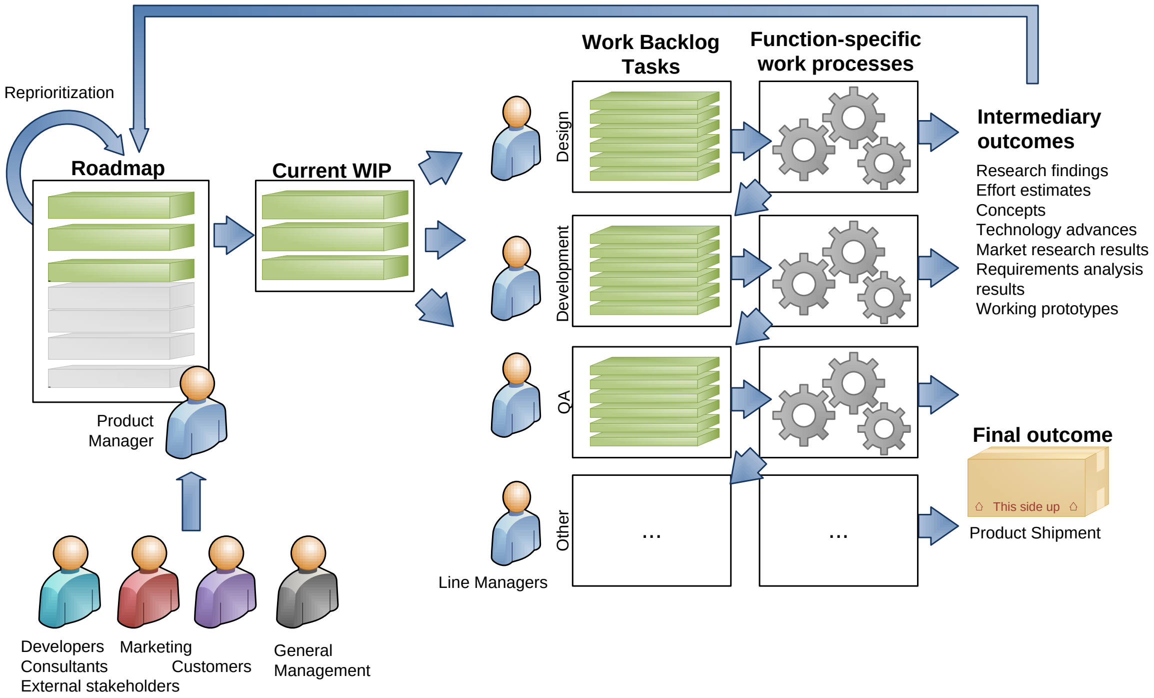


Figure 32 - Product lifecycle management as a global process

The process integrates well with existing process models such as agile software development processes. For example, the development work queue forms the basis for the sprint backlogs in the Scrum process (see Figure 33).

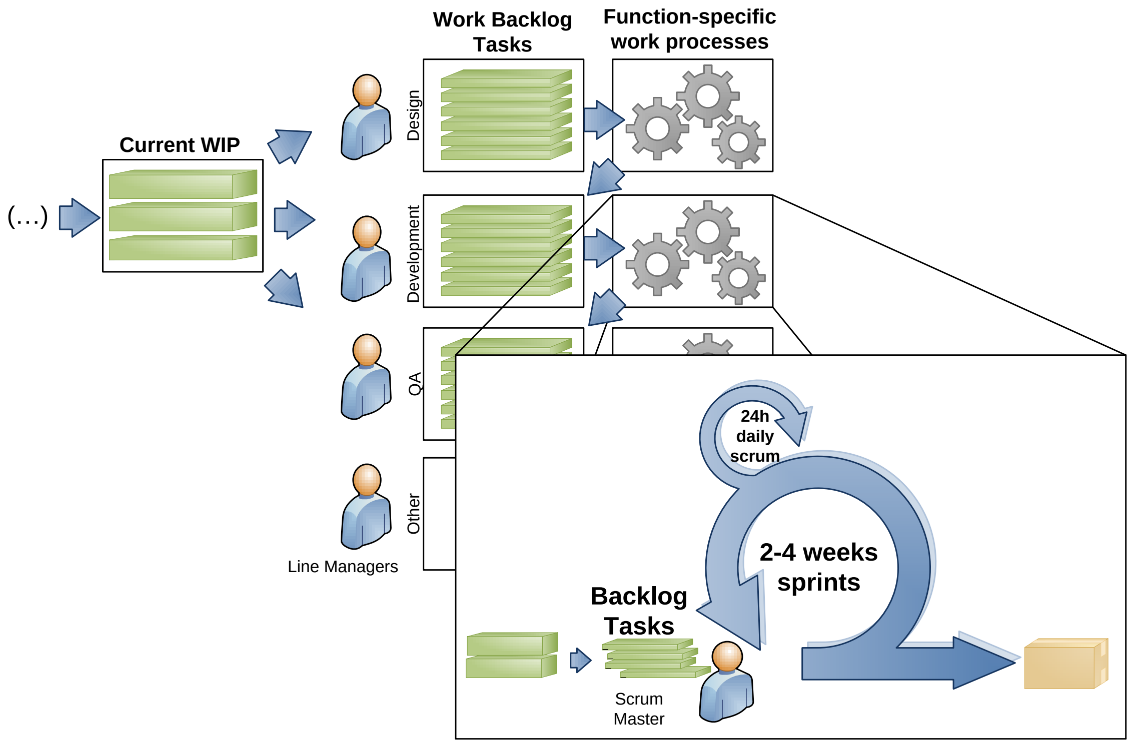


Figure 33 - Integration of Scrum development process into global process framework

The outputs of the design queue are user stories describing the behavior of a particular product feature. In small teams, the design and development work queues might be combined. The same holds true for development and quality assurance. The larger the R&D department grows, the more individual teams and associated work queues need to be managed.

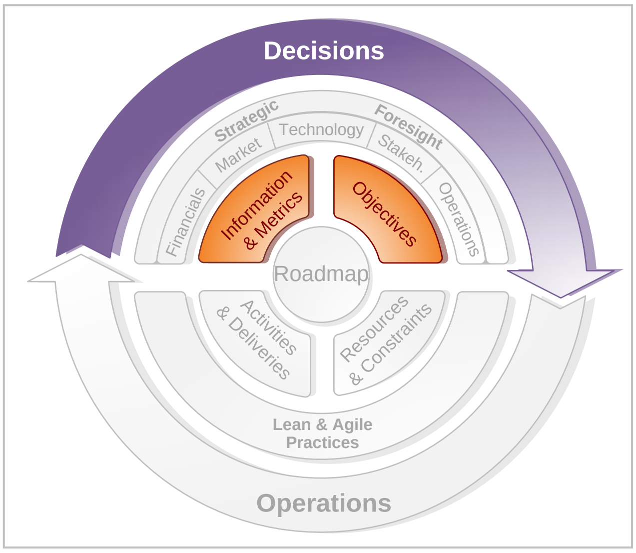
4.6.9.3 Formal and informal roles and responsibilities

With the initial objective of being light-weight and conforming to established processes and roles, typical role models within corporations can easily be mapped to the proposed framework. Product management has the ownership of the roadmap. Thus, product management's role is to drive the roadmap decision process, to collect (and request) relevant information and ensure a transparent decision process. In larger groups, a program manager might be in charge of coordinating the overall task queues and deliveries and make sure the interfaces between the involved departments are functioning. Program management keeps track of the overall status across multiple parallel projects and does the roadmap scheduling. Within the individual groups, ownership should be defined for the subset of work being done within this group. For instance, a product champion in the development team might be responsible for the delivery of the complete software package. Roles originating from formal processes such as the above-named Scrum processes conform to this view. The Scrum Master remains an important driver of the process within the development group and is assigned with additional responsibility to ensure the interfaces with sources of input and consumers of output. The Product Owner's responsibility focuses more on the technical delivery of designed features. Yet, in small teams without separate solution architecture and product design roles, these process steps remain under the Product Owner's responsibility.

Besides formal roles and responsibilities, we do want to refer to a set of informal roles which we see as crucial for a vivid innovation process. Reid et al. [70] studied the fuzzy front-end of innovation, and identified two important role models. Inventors are people taking the role of technology visionaries. Ruminators do market visioning. A combination of these two fosters innovation as an attractive technological invention combined with a market vision. Other researchers pointed out the relevance of champions of innovation (see Schon [78] and Howell et al. [44]), promoters who actively support an innovation and push it through critical stages, particularly early in the process. Witte [100] identified two sorts of promoters, so called power promoters who legitimize and fund projects, set priorities, and get people involved and expert promoters, who actively push the technical solution and solve problems. Later, the theory has been extended by the role of opponents who point out risks and challenges and with that provide a valuable contribution to the innovation process. Gemünden et al. [33] discuss an even broader set of key persons in innovation processes, including technological gatekeepers, process promoters, relationship promoters, power promoters, project leaders and expert promoters and illuminate their interactions. How do these models and observations fit into the proposed framework?
First, we see a need to leave room for "promotion activities". People qualifying for either of these informal roles should be given some leeway to follow-up on opportunities in different directions. Skunk-works projects should not be stopped or artificially structured up to a point where they are sufficiently structured to go through a sophisticated roadmap decision process. This leads to the

previous discussion on strategic buckets for resource investments in research and exploration. Second, with the given model the roadmap is the central focal point up to which an innovation has to manage its way in order to reach the structured NPD process. It should thus be open for inputs from any side like an innovation committee. Still, a more detailed discussion on innovation roles particularly in association with the proposed management and decision framework is beyond the scope of this work and may be subject to a separate research endeavor. We leave the reader at this stage with the notice, that this is an important factor often alleviated in discussions on process models. Awareness on these factors helps to understand the dynamics outside the mere processes and defined workflows and interfaces, which drive innovation to a significant share.

4.7 Innovation Controlling



The last part of the proposed framework is concerned with measuring and continuously improving the effectiveness of all activities throughout the product lifecycle. In literature, both supporters and opponents of structured innovation controlling mechanisms can be found. The ones argue that control increases the efficiency of the process (cf. Davila et al. [29]); others point out that it may harm a firm's innovativeness (cf. Amabile [2]). We hold the opinion that on a high-level perspective, formalized controlling mechanisms and metrics provide orientation and guidance for day-to-day operations. Yet, the meaning and quality of individual indicators and controlling metrics being applied needs to be discussed qualitatively. Relying solely on hard facts and numbers in an environment characterized by high uncertainty might easily lead to wrong assumptions and conclusions.

In search for a suitable innovation controlling framework fitting the requirements of agile product lifecycle management, we reviewed three basic frameworks: the Innovation Scorecard [29], the Innovation Value Chain [39], and Balanced Scorecards [48]. The innovation scorecard model focuses on inputs, processes and outputs. The framework is simple and intuitive, but lacks relevant aspects such as a distinction of inputs from internal, cross-functional or external sources which are considered in the Innovation Value Chain model. Scorecards on the other hand implicitly cover aspects of the roadmap decision process and again apply different perspectives on control (a financial perspective, a customer perspective, an innovation and learning perspective and an internal business perspective). Given this overlap with other aspects already covered in the framework, we believe the model introduces additional overhead, and thus propose to apply the Innovation Value Chain model originating from Hansen and Birkinshaw [39]. The framework is illustrated in Figure 34.

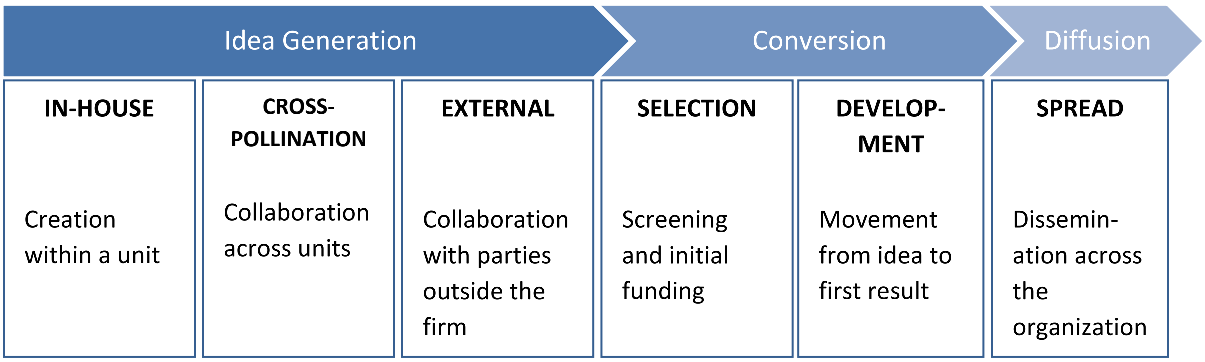


Figure 34 - Innovation Value Chain

Source: Hansen and Birkinshaw, 2007 [39].

Along this value chain we defined a list of concrete key performance indicators (KPI) for the software industry. A listing of KPIs used in the course of this work can be found in the appendix, section 7.5.

Next to the innovation controlling framework, we propose to apply value stream mapping (cf. Cohen [21]) to measure process efficiency and cycle time. As indicated in Figure 35, the aim is to identify and eliminate idle times in the NPD process.

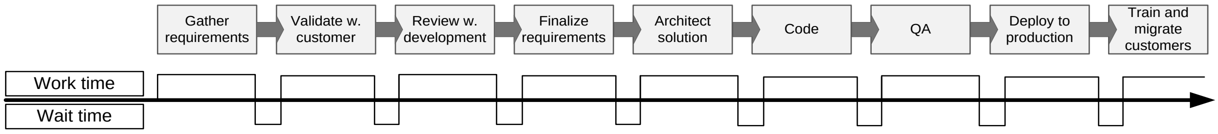


Figure 35 - Value Stream Mapping

Source: Cohen, 2010 [21]

In order to apply the model, the relevant work and wait times have to be tracked and a ratio of wait time to work time is computed. Too long wait times indicate a too large number of work in progress items or process bottlenecks.

Overall, the proposed frameworks (innovation value chain and value stream mapping) point out the most relevant areas for controlling and monitoring. Within these areas, relevant metrics have to be selected matching the individual needs of a firm. The metrics should be aligned with the decision perspectives and criteria, and overall strategic goals. For instance, when dedicating a strategic resource bucket to new products, the relative return on these particular products, their diffusion and acceptance should be measured in order to able to judge whether the bucket is effective, should be increased or decreased. In this way, a seamless interplay of decisions, operations and controlling with continuous feedback loops is possible. Finally, also the incentive system needs to be aligned with controlling metrics for setting measurable and clear goals.

Abstract. The evaluation based on a practical implementation and problem-centric interviews showed that ranking is a powerful tool to enforce, communicate and execute clearly defined priorities. The decision process is transparent and justifiable. The impacts of roadmap decisions are instantly rendered visible. In operations, sophisticated software support is required to implement the model in a broader context. Product modularity, instant testability, resource versatility, divisibility of features, a malleable medium and a flexible distribution channel are prerequisites for applying the framework.

5 EVALUATION

5.1 Evaluation method

An initial evaluation of the proposed method has been carried out in two steps. First, selected parts of the model have been implemented in practice at the software firm where the initial field study was carried out, and results and implications were observed. The implementation included the modeling of skills and resource matrices as described for the roadmap scheduling, the use of a ranking for projects on the roadmap instead of the prior use of an ordinal scale of priorities, and the implementation of ranked work queues for different teams (product design and product development) over the period of two months. Second, the full framework and particularly the items not implemented thus far were presented to managers of all groups involved in the NPD process and a representative of the executive management team to discuss relevance, perceived benefits, weaknesses and applicability.

5.2 Results and findings

5.2.1 Roadmap management and decision process

The first change implemented during the evaluation phase was the introduction of a project ranking instead of an ordinal scale of priorities. Initially the current roadmap, which was managed as a list of projects assigned with priorities (critical, very high, high, medium, low), was converted to a ranked list. Projects with equal priorities were sorted into a ranked order by product management. No additional criteria were applied initially for establishing the ranking. We observed that it was done to a large extent based on tacit knowledge implicitly already present, and that product managers already had a mental model for ranking the project, or at least preferences to do so. Mainly the same criteria were applied to argue for the ranking as were to do the prioritization to priority levels. When presenting the ranked roadmap to the group of line managers involved with design, implementation and testing, the ranking raised several questions for discussion, mainly, to justify why certain projects are preferred over others. Ultimately, feedback of stakeholders was positive on the ranking as priorities are clearer and it provides indication for which tasks to do first. From our observations we conclude that establishing a ranking on the product roadmap is a powerful tool for several reasons. First, it stimulates discussions and requires stakeholders to really agree on priorities. While on ordinal scales there is always the possibility of compromises by assigning two projects with the same priority, the ranking forces a clear decision at planning level. Yet, it requires strict rules for the ranking which do not allow an equal rank of two items. Second, it eases the communication and justification of set priorities. It is much easier to argue why a certain project ranks higher than another, rather than to argue why it was assigned with the priority level "very high", instead of "high" or any other priority. Third, it sets the ground for operations by clearly indicating what is to be done first. It leaves less leeway for operations to "interpret" priorities, and is thus more suitable for a guaranteeing that all stakeholders do have the same understanding of priorities. Finally, the ranking is well-suited to communicate with other stakeholders such as the board of directions. We observed a particular interest in the question of what is currently done, and what is next. Both aspects are expressed transparently in the ranking.

We observed difficulties on the question of whether to establish one global ranking, or several rankings either by strategic buckets of resource investments or logical feature groups. One global ranking turned out to be difficult to establish in case of projects being very different in nature. One executive commented that it was like comparing apples and pears. We recommended in that case splitting up the ranking pool in different groups, where a certain amount of resources will be dedicated to. For instance, when deciding to invest 20% of resources into the strategic bucket "usability enhancements for existing customers", than a ranking for usability enhancements only is established.

Additional techniques and decision analysis aspects proposed to establish the ranking were discussed with stakeholders. Generally, the biggest perceived advantage named in interviews was the aspect of justification of decisions and actions. The decision perspectives and models for input help arguing for a particular decision and pointing out which criteria were applied, how the information was aggregated and how the final decision was done. Furthermore, the approach was judged as transparent. All management is done around the ranked list of projects, decisions are immediately visible when changing the ranking, and transparently populated to all stakeholders.

The feedback on the formal decision process itself was mixed. Mainly, concerns about complexity and effort have been expressed when having to deal with a large number of pair-wise comparisons. From a theoretical point of view, the process works and is effective, but for a practical implementation, at least structured and effective tool support is necessary to manage the decision process and track all decisions and comparisons. Several software systems exist to manage such a decision process. Yet, it requires integration with the full tool chain to be managed effectively and avoid having to administrate projects separately in different tools. In contrast, the informal decision approach was described as effective based on proposed checklists and questions for orientation. At the same time, concerns were uttered that such roadmap committee meetings turn into lengthy discussions if argumentation is not underlined by facts researched prior to this meeting. We generally believe the success of the ranking process will depend in that case strongly on the preparation of participants. The informal approach might also be subject to a larger number of decision biases. Particularly when done in a cross-functional team meeting, capabilities of arguing for or against a particular project impact the overall process. Different stakeholders promote different projects and try to push them to the highest possible rank. Participants with a rather aggressive style of arguing might be more successful in promoting their proposal or blocking others' projects. We thus see a strong need for neutral and unbiased moderation and an effective communication process and rules.

Representatives from executive management appreciated the transparency over resource utilization via strategic buckets, and the possibility to steer and direct efforts by adjusting these buckets. We support this view from the perspective that these buckets really help to align high-level strategic planning with the more fine-grained roadmap, but see a danger in "over-steering" the buckets. In practice, there is only limited flexibility to change such buckets within a short time period. Operations and resources need to be realigned after each change. Another bucket might require different resources and different skillsets, which cannot be acquired instantly. The same holds true if resources become obsolete.

The decision perspectives and proposed frameworks such as using the Kano model or QFD were considered beneficial as additional aspects can be taken into consideration effectively, which again contributes to the first and foremost benefit perceived: to make justifiable decisions. Literally, managers like to "tick off" check lists and make sure to have them considered and to have done the decision process thoroughly.

The evaluation furthermore confirmed that the proposed concept for combining effort estimated with target scoping is a powerful control mechanism which satisfies interests of both technicians and marketing. Technicians have planning security for the basic features and can start off with implementation and design of these core features instantly. Marketing receives a flexible buffer to add features with strong performance attributes or excitement attributes throughout the project as the requirements become more clear and potentially after having initial customer feedback on prototypes or beta releases, as long as these extensions are within the target scope and costs. Yet, it will require the implementation of several long-term projects based on this model to really judge strengths, weaknesses and potential pitfalls.

5.2.2 Workflow and operations

In terms of the proposed operations model and workflow, the aspect of managing work in priority queues for different departments was implemented and tested with the product design team and one of the development teams. The concerned product development team, consisting of ten team members was operating based on the agile Scrum software development process. Thus, the team was used to planning based on a ranked backlog of tasks, which are estimated and assigned to team members in sprint cycles of two to three weeks. In order to implement the product lifecycle management system, the project backlog was extended, and developer tasks were defined as sub-tasks of the overall enhancement project, which did not imply any significant behavioral changes for the team. In contrast, the product design team previously did not separately manage design tasks for a project. The creation of the functional descriptions was one monolithic task / phase in the project plan. Individual aspects were managed in personal to-do-lists of the team members. In order to implement the framework, an issue and task tracking system used for development was introduced

also for the design team, and design tasks were managed in the system in a ranked queue.
Overall, the transparency achieved with the unification of task management eases the communication between the departments, and it allows overlapping different project phases much stronger. Previously, development waited until the functional description was fully completed by product design. Then it was handed over to development, later changes required a lengthy and bureaucratic change approval process. With the new approach, design tasks were ranked by the priority for development, i.e., what does development need to know first to implement the next step. If a subset of design tasks is finished concerning one particular, modular item, this can already be implemented. This change and the stronger overlap of phases significantly reduces the cycle time of a project and confirms the prior work on fast and flexible product development (see Wheelwright and Clark [99], Smith and Reinertsen [81][82], Iansiti [46] and MacCormack et al. [60]), which argues that overlapping phases are the key to speeding up time-to-market.
One weakness we observed during the relatively short implementation time was a lack of resource balancing across the departmental priority queues. While within teams, load-leveling happens implicitly by the assignment of tasks to different team members, little possibility exists for load-leveling across different teams. If a bottleneck exists in one queue due to resource shortages, the whole chain suffers. This phenomenon is not exclusive to the proposed framework, but the framework appears to emphasize these bottlenecks as tasks and dependencies are coordinated on a more fine-grained level and small delays appear to have a stronger visible impact. This observation confirms the high transparency of the approach and how information is automatically populated and has impact but we see a danger of tensions between teams if the load is not balanced and there are perceived bottlenecks on either side.

5.2.3 Resource planning and roadmap scheduling

The roadmap I/O model, when presented to stakeholders in product development and marketing, clearly was clearly seen as one of the core benefits, if it can be implemented as discussed. From a theoretical perspective, being able to transparently point out resource constraints and the impact of roadmap decisions on schedules is important for both people involved in operations and people involved in decision making. Yet, concerns were expressed as to which extent this is possible in practice. The implementation and accuracy of the model strongly depends on how precisely resource compatibility and skills can be reflected in the planning process. The collection and regular update of these data means significant effort for involved managers. During the implementation phase, we collected resource compatibility and skills in a spreadsheet and manually carried out the scheduling. This quickly turned out to be insufficient for a large number of involved people, projects and constraints and a longer planning horizon. The roadmap I/O could not be implemented without algorithmic support at all. The evaluation clearly showed that this aspect of the model requires sophisticated software support. The system applied should integrate the project ranking with the resources and skills matrices and the proposed scheduling and optimization approach to immediately derive a schedule based on all modeled parameters when defining or changing the ranking. Additionally, we conclude from the experiences with the initial implementation that it is vital to keep the granularity at which skills are defined rather coarse to avoid overwhelming complexity due to a too high number of constraints.
Furthermore, in order to do more accurate planning, for instance of on-costs and post-launch

resource utilization, more sophisticated and extensive reference numbers are required. These could be derived either from precise internal records, or from industry standards. Though, these are still to be determined and are currently not documented to our best knowledge.

Generally, the aspect of information sharing is vital for the overall framework. It requires a “central wall” of information, accessible for everyone involved with the process at any stage. This central piece of information should clearly show priorities, schedules and responsibilities.

5.3 Method adoption

We consider the proposed framework a potential process innovation. In order to evaluate its effectiveness and potential for broad acceptance and diffusion, we applied a list of criteria proposed by Rogers [72]. These criteria are important factors impacting the adoption of an innovation. They include the product's/technique’s perceived advantage or benefit, the riskiness of a purchase or method introduction, the ease of product use or method application, i.e., the complexity of the product/method, the immediacy of benefits, observability, trialability and the extent of required behavioral changes. Results are summarized in Table 12.

Criterion	Findings
Perceived advantages	▪ Justifiable and well-documented decision process. ▪ Everyone is on the same page: priorities are clear and easy to communicate and understand. ▪ Impact of roadmap decisions is transparent and explicit. ▪ Well-managed global workflow reduces idle times in the process and shortens time-to-success.
Riskiness	+ Limited operational risk when implemented as a process on top of existing operations. - High risk in terms of criticality for an enterprise: false decisions on the product roadmap may significantly threaten a firm.
Complexity	+ Simple ranking process which can be detailed on demand. - Management of resources and skills perceived as tedious. Software support required for implementing roadmap scheduling and roadmap I/O.
Immediacy of benefits	+ Immediately visible changes in planning process and communication. - No immediately visible benefits in terms of project cycle time reduction – requires full process implementation and training, and a seamless operation of the model for a new product release.
Observability	Limited – Needs either to be deployed for trying the model, or observability might be given through peer groups having collected experiences.
Trialability	+ Trialability is given for the decision making and prioritization process - it can be tried out with low effort to compare the output with currently used planning method.

	-	Trialability is not given for the roadmap I/O concept as it requires software support to be done effectively. Such systems are not easy to integrate with the existing tool chain.
Required behavioral changes	+	Little change required in terms of global workflows. The method integrates well with existing and established processes and tools. Method strengthens the decision process without introducing high additional effort by restructuring and aligning it.
	-	Significant behavioral changes and change of mindset required in terms of planning process implied by combining predictive with adaptive paradigm.

Table 12 - Evaluation of method according to Rogers' criteria

We conclude from the evaluation that for a broad adoption of the full model, it will require a growing body of knowledge and best practices around the core methodologies, integrated and easy-to-use software support and leading examples of successful implementations. In a first step, parts of the method such as the ranking process and the associated decision analysis techniques can be adopted before implementing the operations model to fully leverage the models' potential benefits.

5.4 Limitations and general applicability

Thus far, the proposed framework has been developed and evaluated only in the context of software development. Due to frequent change and a highly dynamic environment, it has been an attractive sample to conduct the initial research in the field. Yet, similar challenges in product lifecycle management are encountered in a large number of other industries as well. In order to judge as to which extent the model can be generalized and applied in a different context outside the software space, we identified certain key characteristics of software development, upon which the model relies. We claim that for an environment showing similar characteristics, the model might potentially be suitable and applicable. The list is completed by related findings of Smith [83] on the applicability of flexible product development outside the software space.

Modularization and isolation

In the software domain, object-oriented design has contributed significantly to modularization of products and the isolation of independent product components [83]. Having independent and isolated modules allows implementing, changing and testing them independently, and more importantly, in parallel. Modularity eases the flexible distribution of tasks and dynamic scoping for adaptive planning.

Core platform and satellite features

Strongly linked with the aspect of modularity, many software systems do have a stable core framework which rarely changes, such as a backend engine, the core of an operating system and the like. On top of this core system, various features are implemented such as different user interfaces or user functions and processes. These satellite parts make use of the core platform and their implementation effort is relatively small in relation to the full system. Thus, they can be exchanged and the software can be reshaped flexibly according to customer needs. Such a structure generally gives planning flexibility and the proposed framework takes this aspect into consideration, for instance be balancing resource investments between core and satellite parts.

Divisibility of features

In many software systems, product features are rather independent, which allows developing one feature after another (or in parallel) with little interdependencies. This means that scope changes only impact the added or removed features, but do not require significant changes to what was already developed. In contrast, in many industries adding a product feature may require redesigning the complete product, the use of different material or other significant change. Divisibility of features and associated tasks clearly is an enabler for the implementation of the proposed approach. Otherwise, the cost of late change increases exponentially.

Malleability of the medium

Smith [83] argues that in general, the malleability of the software medium makes change relatively easy. In contrast to other industries where a change in the product design might require changes in machinery or production facilities, a software system can be changed without impacting the production environment. Out of this consideration, the proposed framework does not take into consideration any costs of change associated with a design decision and applies only in a context of a flexible medium.

Channel flexibility and distribution of frequent change

The software industry has developed mechanisms to frequently and cheaply distribute changes and enhancements via internet downloads, software migration and update routines, online testing and sandbox environments, or most recently software-as-a-service offers where the creators of the software manage their environment by themselves and can frequently deliver updates to all users. This channel flexibility gives access to instant customer feedback and allows leveraging the short planning cycles and frequent deliveries.

Instant assembly and testability

In the software industry, so-called continuous integration and automated testing are a general best practice. It means that an integrated and testable version of the product is built automatically at a regular basis. Continuous integration servers create nightly software builds that can be tested on the next day. Software further allows automating the testing of individual software units and the whole system. In the light of frequent change this allows to check system integrity with every product modification.

Trialability and access to customers

Smith [83] claims that for IT projects, it is relatively easy to find customers that can be involved in the development. In order to leverage the planning flexibility, we see this as a key aspect for gathering and incorporating early feedback.

High uncertainty and emergent requirements

The model has been designed for environments and markets characterized by high uncertainty, risk, and emergent requirements. In a very stable environment, purely predictive and plan-driven approaches will most likely show better results and efficiency.

High design flexibility and choice

A sophisticated decision process is only required if there is design flexibility and choice in both product directions and project proposals. In the software industry, a problem can be addressed using very different approaches, from specific to generic solutions. The same holds true for other industries. The proposed approach applies only in a context with sufficient choices to be made. If the environment, the products, processes and designs are mostly set, there is little room for agility.

Versatility of resources

The operations model and particularly the activity scheduling approach are designed for an environment with resource versatility, i.e., resources can be assigned flexibly to different projects and tasks depending on the current strategic focus. We see resource versatility as a general strategic goal for corporations operating in a dynamic market and it is a prerequisite for the presented framework.

6 SUMMARY AND CONCLUSIONS

In this thesis, a framework for agile product lifecycle management has been proposed. It is built upon the idea of establishing and constantly updating a ranked list of projects on the product roadmap and provides guidance for decision making to establish said ranking, and the operational execution. We discussed how different decision perspectives are taken into consideration, including financial valuation, market factors, stakeholders' interests, technologies and operational constraints, as well as how to aggregate these using the analytical hierarchy process. Principles of lean production are applied to manage the global workflow throughout the product lifecycle.

The framework addresses many of the challenges inherent to product lifecycle management in a dynamic market environment. Different driving forces impacting roadmap decisions are laid out in a transparent decision process. Uncertainty is addressed with incremental information gathering, a combination of predictive and adaptive planning and frequent iterations to learn and fail early. The framework applies to environments characterized by high uncertainty, with modular product architectures and divisible features, versatile resources and a fast and flexible distribution channel.

An initial evaluation of the method confirmed the effectiveness of the overall approach, particularly the consistent use of ranking to manage priorities, the combination of predictive planning based on estimates and adaptive planning based on target costing, the decision process and the global workflow. Still, there is much room for future research based on the initial results. A broader study on total costs accrued in different lifecycle stages, as well as other indicators applied in the framework is required to generalize the reference model and provide more concise guidance based on industry-average numbers. An evaluation of long-term results of the method across different firms and industries is needed to confirm its general applicability. Other interesting research questions are to compare the results of different decision analysis methods applied to roadmap decisions, or to evaluate which organizational design favors the framework implementation (cf. Westerman et al. [98]).

7 APPENDIX

7.1 Summary of basic NPD models

Saren [77] identified seven distinct categories for NPD processes, summarized and detailed by Trott [90]:

In **departmental-stage models**, each department performs certain tasks in a linear model and hands over to the next department after completion. This “over-the-wall” approach today is generally seen as outdated as it lacks communication and coordination among different stakeholders.

Cross-functional models are based on the idea of assembling cross-functional NPD teams and fostering collaboration and knowledge sharing. Yet, neither is this “easy solution” always possible and manageable, e.g. in case of multiple parallel NPD projects sharing joint resources, nor does it provide much guidance on how the cross-functional team can work most efficiently to create new products. Thus, it is rather a tool usable in any process model, but not a comprehensive model by itself.

Decision-stage models view NPD as a series of decisions to be taken.

Conversion-process models see NPD as a function of input into a “black box”, and conversion into a product. Few details and research exist on both categories. Thus, their practical relevance appears negligible.

Response models rely on the behaviorists’ approach of change analysis and focus on the response to a new product proposal or idea. Again lacking concrete implementation guidance, these can be considered as agile in a sense of responding to input signals and reacting to these by adapting the process.

Activity-stage models, as their name suggests, focus on individual activities rather than functions, and allow for iteration and feedback loop. While still being criticized as a different shape of “over-the-wall” model [90], one of the most wide-spread NPD process models is classified into this category, i.e., the Stage-Gate© model proposed by Cooper [22]. It divides the NPD processes into six subsequent stages from discovery to product launch with intermediary decision gates allowing discarding or continuing a project. Such models are able to reduce project risks by an incremental lock-in of resources into a project and the systematic decision making processes.

Finally, the most recent models for NPD are **network models**. These models are built on the idea of constant knowledge accumulation from various inputs, either from internal or external sources. A network of external linkages is coupled seamlessly with internal development. Figure 36 illustrates these models from a high-level perspective.

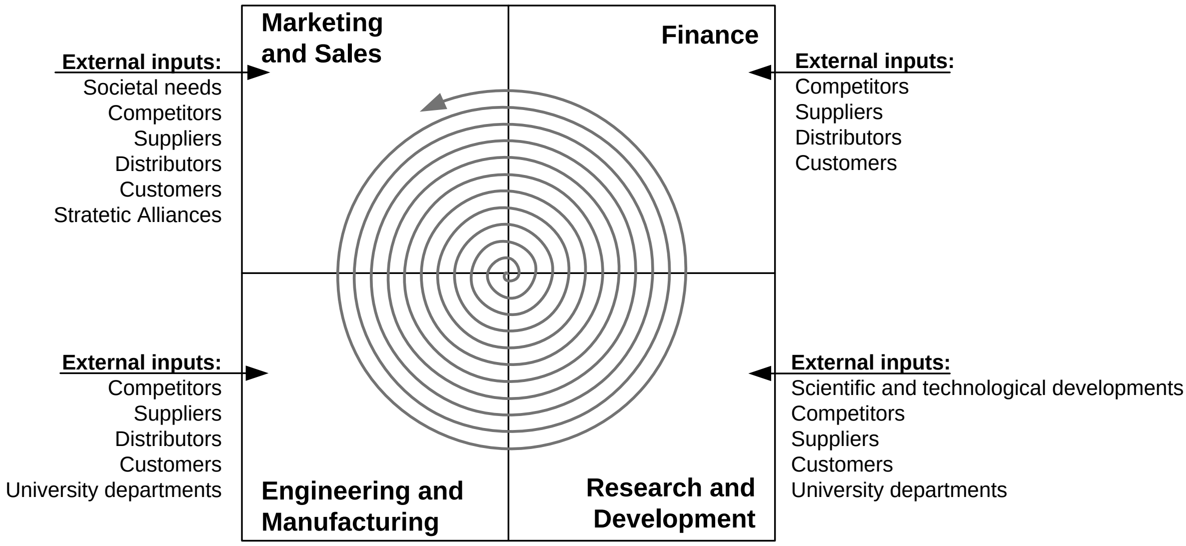


Figure 36 - Network model for NPD

Source: Paul Trott: Innovation Management and New Product Development [90].

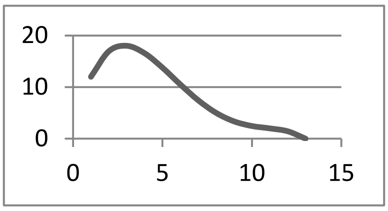
7.2 Maintenance efforts reference model

Based on the findings from the data analysis, a classification of product features into different maintenance classes is proposed. The classes are characterized by two dimensions:

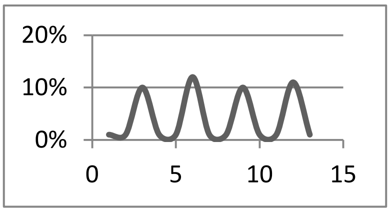
- Total maintenance effort over time
- Correlation of maintenance effort with number of product items sold

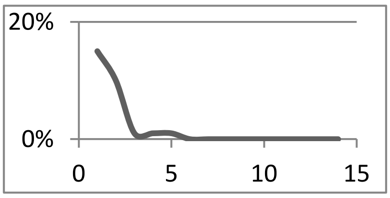
The first dimension describes the maintenance curve based on a fixed number of product items in use at a fixed number of customers, whereas the second dimension describes the changes of this curve in relation to the quantity of product items in use.

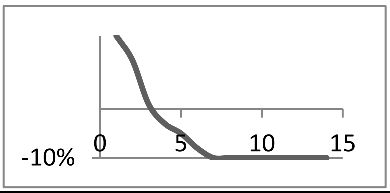
7.2.1 Total maintenance effort over time

<table>
<tr><th>Maintenance curve</th><th>Description</th></tr>
<tr><td>


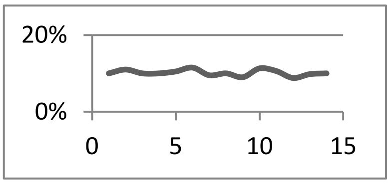
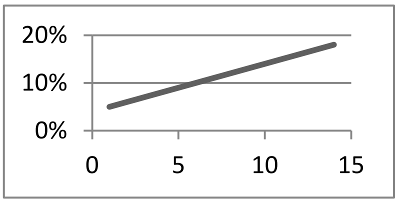
</td><td>Initial peak – steady decrease
This pattern is characterized by an initially high maintenance effort, increasing after the launch, and then constantly decreasing.

For software components, this pattern can be observed with new modules and user interfaces, where most feedback is collected and incorporated after initial deployment at the first customers.</td></tr>
<tr><td>


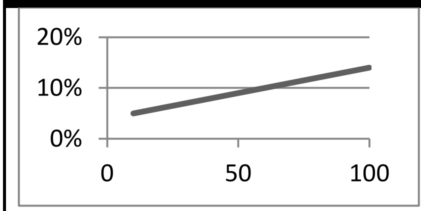
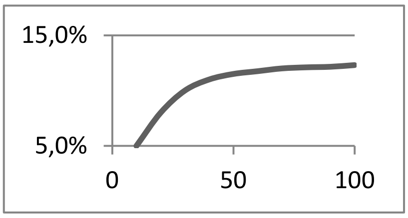
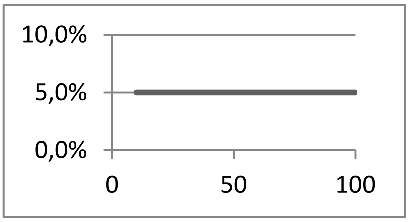
</td><td>Regular/Irregular peaks
This pattern occurs for software components, which are either in use only at certain times (e.g. end of quarter/year), or which are heavily dependent on an external component, and require maintenance if this component changes.</td></tr>
<tr><td>


</td><td>Initial effort
This pattern occurs, if a component requires maintenance efforts when being rolled-out, but does not cause any further maintenance effort after being deployed.</td></tr>
<tr><td>


</td><td>Effort reduction
This pattern is impossible to measure in the maintenance data. Yet, there are components/features, which decrease the total maintenance efforts for a product after introduction. Particularly, this concerns any usability enhancements,</td></tr>
</table>

<table>
<tr><td></td><td>performance/reliability enhancements etc.</td></tr>
<tr><td>
</td><td>Constant effort
Certain components result in a constant maintenance effort over their lifecycle. For instance, these might be components with a planned and regular maintenance cycle. Yet, in practice not much evidence for this pattern has been found.</td></tr>
<tr><td>
</td><td>Steady increase
Evidence has been found that even a steady increase in maintenance efforts for a certain component is possible throughout its lifecycle.
Possible explanations might be bad design making later maintenance more costly, a growing feature set or growing complexity. This is a clear indication for the need of refactoring, usability enhancements and simplification.</td></tr>
</table>

7.2.2 Correlation of maintenance efforts with number of product items in use

<table>
<tr><th>Correlation</th><th>Description</th></tr>
<tr><td>
</td><td>Linear
A linear correlation between maintenance effort and number of licenses sold indicates fixed variable maintenance costs without economies of scale. This pattern is unlikely to occur in the software industry and has not been observed in the data.</td></tr>
<tr><td>
</td><td>Logarithmic
This pattern has been observed most frequently. The maintenance effort increases with the number of deployed licenses, but the correlation is logarithmic, i.e., from a certain point almost no additional effort stems from further deployed licenses.
This can be explained from the fast that the number of defects/sources of maintenance efforts is limited and will be mostly covered after the rollout at a critical mass of customers.</td></tr>
<tr><td>
</td><td>None
For certain components there is no increase in maintenance effort with more licenses being sold. Potentially, these are very simple components or enhancements which even reduce maintenance effort.</td></tr>
</table>

7.3 Real-options analysis example based on Black-Scholes formula

In order to evaluate results of real-options valuation based on the Black-Scholes formula, a real-world example is considered: A firm offers a palette of software tools, each working independently, managing independent data and communicating with each other via different integration mechanisms. If data is exchanged between the tools, it has to be replicated. Yet, the architecture is sufficient for current installations where the tools are sold mostly independently. A potential roadmap feature (project 1) is to establish a central data store and a message-based integration layer between the tools. The implementation effort is estimated to be 2000 man days (MD) plus efforts of 100 MD in year one for rollout. How can a business case for this feature be constructed?

Assumptions:

- In the current sales channels, the feature will not yield any additional revenues.
- The feature will simplify installation, and reduce maintenance efforts by 50 MD per year after year one for a calculation period of 5 years.
- The feature enables the later integration of another software product to the product palette. The following table summarizes these integration options. The expected cash flows from this integration (project 2) are listed in Table 14.
- The costs of 1 MD are estimated to be € 1.000,00 in the example, the WACC is assumed to be 20%, and a current risk-free rate of 10% is used as the basis for calculations.

Based on these assumptions, computing the net present value (NPV) for the original project (without taking options into account), not surprisingly yields a negative result. Table 13 shows the projected cash flows, whereby only the reduction in maintenance efforts can be taken as a basis for the calculation.

Year	0	1	2	3	4	5
Sales potential	-	-	-	-	-	-
-Dev. & maintenance effort		-50.000	-50.000	-50.000	-50.000	-50.000
Net operating profit	-	50.000	50.000	50.000	50.000	50.000
-Capital investments (initial outlay)	1.000.000	-	-	-	-	-
Cash flows	-1.000.000	50.000	50.000	50.000	50.000	50.000

Table 13 - Example for projected cash flows from maintenance reduction

The NPV of the project is € -850.469,39. Thus, without taking the associated options into consideration, the project will not pass any structured roadmap decision process based on a negative financial valuation.

In order to take the option of integrating the other software product into consideration, its cash flows have to be projected as well (see Table 14 for the figures used in the example).

Year	0	1	2	3	4	5
Sales potential	-	-		1.000.000	1.500.000	2.000.000
-Dev. & maintenance effort	-	-		200.000	220.000	250.000
Net operating profit	-	-		800.000	1.280.000	1.750.000
-Capital investments (initial outlay)	-	-	1.300.000	-	-	-
Cash flows	-	-	-1.300.000	800.000	1.280.000	1.750.000

Table 14 - Example for projected cash flows from software project

The project has a present value (PV) of € 2.568.287,04 in year 2, which is equivalent to a present value of € 2.122.551,27 in year 0 when discounted based on the defined risk-free rate. Yet, these are not actual cash flows. These are theoretical cash flows associated with an option enabled by the initial project. In a next step, the Black-Scholes formula is applied to calculate the value of this option. As these cash flows are highly uncertain, the standard deviation of comparable high-tech stocks is taken for reference to account for the uncertainty. We will calculate the option value based on a standard deviation of 35%. For step-by-step explanations on the calculations done below, please refer to Brealey et al. [15].

$$PV(excercise\ price) = \frac{1.300.000}{(1.1)^2} = 1.074.380{,}17$$

$$Call\ value = [N(d_1) * P] - [N(d_2) * PV(EX)]$$

$$d_1 = \frac{\log(\frac{P}{PV(EX)})}{\sigma * \sqrt{t}} + \sigma * \sqrt{\frac{t}{2}} = 1{,}426$$

$$d_2 = d_1 - \sigma\sqrt{t} = 0{,}820$$

$$Call\ value = 0{,}923 * 2.122.551{,}3 - 0{,}794 * 1.074.380{,}17 = \mathbf{1.106.379{,}49}$$

As can be seen from the result, despite the high level of uncertainty accounted for with the standard deviation, the option on project 2 has a call value of 1,1 mln., which compensates for the negative net present value of the project 1. By considering the options value, the overall NPV turns positive.

7.4 Indicators for the S-curve

Indicators for…	Measure	Discussion
Product quality and maturity	Number of critical, major, minor, and trivial bugs reported per unit of time	This metric is relevant not only for continuous evaluation of product quality, but also indicates maturity: the higher the ratio of minor and trivial issues reported, the more mature the software in terms of the existing feature set. This information needs to be combined other indicators to see how mature and complete this existing feature set is.
	Frequency of change requests per customer/user	Indicates how complete the feature set is and whether it (still) fulfills customers' requirements. If the number is decreasing, the product is more mature, if it starts increasing, customer needs and expectations might have changed over time or the product is generally immature.
	Number of small vs. large-scope feature requests	The type of requests from existing users is also relevant for determining maturity: generally, the larger the scope of the requests the less mature the product offering. If the offering is rather complete, only minor changes are requests.
	Number of support cases per product, per KLOC (kilo lines of code), and per time period	This metric adds to the quality aspect, but covers also an indication of usability and manageability of the product.
	Rate of improvements in product performance attributes per time unit (speed, throughput, scalability, amount of data, usability metrics, etc.)	Any measurable product performance attribute might be used as an indication for product maturity. The relevant question to ask is: how much more is possible in the given product architecture?
Market potential	Product sales per time unit	Despite its relevance, this indicator needs to be used with care due to the dependent variable of sales performance. The figure needs to be clearly attributed to either product or sales success/failure; otherwise it cannot be applied in this context.
	Revenue share from core product	Many innovative firms reported applying this metric in goal setting for innovation. We believe it can be used also as an indicator for saturated markets if given the same product offering, revenue shifts towards the newer products.
	Number of new-to-the	Indicates the potential for new customers. If

	product users vs. users switching from competitive products	customers are acquired mostly from competitors,
	Position in technology cycles and analysts' reports	This factor has been mentioned as a "sentiment" indicator for a product category. Despite the fact that many of such reports, technology hype cycles etc. rely on punctual research, it has measure impact on buyers' behavior and is nevertheless relevant.
	Number of deals lost per time unit with the product (for performance reasons)	This is a simple, yet relevant indicator for competitive position and strength of the product, and thus also relative product maturity.
	Major selling points in closed deals	Not a numeric metric, this might still indicate the reasons why a product is bought, which allows to draw conclusions on technology maturity.
	Trends in demand	Indicators on new leads generated with the product, requests/inquiries etc. might also be considered in this context.

Table 15 - Examples for indicators to measure product maturity

7.5 Innovation controlling metrics

Stage	Aspect	Indicators
IDEA GENERATION	In-House	▪ #ideas proposed (new to firm | new to industry) ▪ #ideas reviewed and funded ▪ #new products in pipeline ▪ #products started ▪ #"high-profile" employees, e.g. with research background ***"Have a constant stream of valuable incremental innovations to keep market lead in the segment"***
	Cross-Pollination	▪ #proposals tracked in cross-functional design workshops ▪ Hours spend in cross-functional interaction and knowledge exchange ▪ #Contributions to internal wiki/portal system ***"Foster cross-functional exchange of knowledge and ideas"***
	External	▪ #external research projects (master theses, PhDs, university projects) ▪ #research publications ▪ #ideas submitted by partners ▪ #external ideas reviewed and funded ***"Make sure to leverage multiple external sources and foster collaborations with partner firms and universities in particular"***
CONVERSION	Selection	▪ #ideas discarded ▪ Selection rate (#funded ideas/#proposed ideas) ▪ Duration of selection process (from idea submission/project proposal to decision) ***"Make sure to have a fast and efficient selection process"***
	Development	▪ Development costs / revenue (per product line) ▪ Plan vs. actual efforts ▪ Time-to-market ▪ Time-to-beta testing ▪ #Beta customers ▪ #Product revisions ▪ #Bug fixes after rollout ▪ Bugs per KLOC (thousand lines of code)

		▪ Total effort / cycle time ▪ Percent first design meets needs ▪ Team assessment of design effectiveness ▪ #Internal IT impediments ▪ #Process impediments reported ▪ #Time spent on products relative to processes and infrastructure ▪ #Bugs reported by internal QA vs. bugs reported by customers ***“Deliver projects as efficiently as possible, with the least possible number of iterations and revisions; Spend time effectively and reduce waste.”***
DIFFUSION	Spread	▪ %revenue from new products (introduced in less than 3 years) ▪ Customer satisfaction – product, services ▪ Market share in newly addressed market ▪ Revenue growth from new products ▪ Conversion rate: ideas to market ▪ #deployed installations for new products ▪ Percent new features in products ▪ Number of products completed ▪ #Design/industrial awards (or comparable recognition ▪ Position in analysts’ reports (e.g. Gartner magic quadrants, Forrester Wave) ***“Bring new products successfully into the market place, gain fast visibility and diffusion”***

Table 16 – Relevant indicators along the innovation value chain in a software company

8 BIBLIOGRAPHY

[1] Akao, Y., 2004. QFD: Quality Function Deployment - Integrating Customer Requirements into Product Design, 1st ed., Productivity Press.

[2] Amabile, T.M., 1998. How to kill creativity. *Harvard Business Review*, United States, Boston.

[3] Ansoff, H.I., 1975. Managing strategic surprise by response to weak signals. *California Management Review* 18 (2), pp. 21–33.

[4] April, A., and Abran, A.: Software Maintenance Management: Evaluation and Continuous Improvement. John Wiley & Sons, Hoboken, New Jersey, 2008.

[5] April, J., Glover, F., and Kelly, J., 2003. Optfolio - A simulation optimization system for project portfolio planning. *In Proceedings of the 2003 Winter Simulation Conference*. New Orleans.

[6] Bardhan, I., Bagchi, S., and Sougstat, R., 2004. A Real Options Approach for Prioritization of a Portfolio of Information Technology Projects: A Case Study of a Utility Company. *In Proceedings of the 37th Hawaii International Conference on System Sciences*.

[7] Bardhan, I., and Kauffman, R. J., 2006. Optimizing an IT Project Portfolio with Time-Wise Interdependencies. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, 8(1), pp. 162-169.

[8] Beck, K., 1999. Extreme Programming Explained: Embrace Change. Addison-Wesley.

[9] Beck, K. et al., 2001. Manifesto for Agile Software Development. Agile Alliance. [online] Available at http://agilemanifesto.org, 2001 [Accessed 20 May 2011].

[10] Biazzo, S., 2009. Flexibility, Structuration, and Simultaneity in New Product Development. *Journal of Product Innovation Management* 26(1), pp.336-353.

[11] Black, F., and Scholes, M., 1973. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* 81, pp.637-654.

[12] Boehm, B., and Turner, R., 2003. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley Publishing Co.

[13] Booz, Allen and Hamilton, 1982. New Products Management for the 1980s. New York.

[14] Brans, J.P., and Vincke, P., 1985. A preference ranking organization method: The PROMETHEE method for MCDM. *Management Science*.

[15] Brealey, R. A., Myers, S. C., and Allen, F., 2008. Principles of Corporate Finance, 9th ed., McGraw Hill.

[16] Bright, J.R., 1970. Evaluating signals of technology change, *Harvard Business Review* 48(1), pp.62–70.

[17] Cagan, M., 2008. Inspired: How to create products customers love. SVPG Press.

[18] Christensen, C.M., 1997. The innovator's dilemma: when new technologies cause great firms to fail, *Harvard Business School Press*, Boston, MA.

[19] Christensen, C.M., 2010. Innovation Killers: How Financial Tools Destroy Your Capacity to Do New Things, Harvard Business Press.

[20] Cockburn, A., 2004. Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley Professional.

[21] Cohen, G., 2010. Agile Excellence for Product Managers. Superstar Press.

[22] Cooper, R. G., 1993. Winning at New Products, 2nd ed. Addison-Wesley Publishing Co.

[23] Cooper, R. G., 1996. Overhauling the New Product Process. *Industrial Marketing Management* 25(6), pp.465-482.

[24] Cooper, R. G., 2008. Perspective: The Stage-Gate® Idea-to-Launch Process - Update, What's New, and NexGen Systems. *The Journal of Product Innovation Management* 25(3), pp.213-225.

[25] Crawford, M., and Di Benedetto, A., 2008.New Products Management, 9th ed. McGraw-Hill, New York.

[26] Daheim, C., 2009. Corporate foresight. How to organize, run and manage a corporate foresight exercise. Examples and experiences. UNIDO Workshop [online] Available at https://www.unido.org/foresight/rwp/dokums_pres/daheim_cf_presentation_242.pdf [Accesses 18 June 2011].

[27] Daheim, D., and Uerz, G., 2008. Corporate foresight in Europe: from trend based logics to open foresight. *Technology Analysis & Strategic Management* 20(3), pp.321–336.

[28] Damodaran, A., 2002. The Promise and Peril of Real Options. Working Paper, Stern School of Business [online] Available at http://pages.stern.nyu.edu/~adamodar/pdfiles/papers/realopt.pdf [Accessed 29 May 2011].

[29] Davila, A., Epstein, M., and Shelton, R., 2006. Making Innovation Work. How to Manage It, Measure It, and Profit from It. Wharton School Publishing, New Jersey.

[30] Dröschel, W., Heuser, W., and Midderhoff, R., 1998. Inkrementelle und objektorientierte Vorgehensweisen mit dem V-Modell 97, Oldenburg, Munich.

[31] Faulkner, T., 1996. Applying 'Options Thinking' to R&D Valuation, *Research-Technology Management*, pp.50-57.

[32] Furugaki, K., Takagi, T., Sakata A., and Okayama, D., 2007. Innovation in Software Development Process by Introducing Toyota Production System, *Fujitsu Science and Technology Journal* 43(1), pp.139-150.

[33] Gemünden, H. G., Salomo, S., and Hoelzle, K., 2007. Role Models for Radical Innovations in Times of Open Innovation. *Creativity and Innovation Management* 16(4), pp.408-421.

[34] Goffin, K., Herstatt, C., and Michell, R., 2009. Innovationsmanagement: Strategien und effektive Umsetzung von Innovationsprozessen nach dem Penthalon-Prinzip. FinanzBuch Verlag, München.

[35] Green, P. E., and Srinivasan, V., 1978. Conjoint Analysis in Consumer Research: Issues and Outlook. *The Journal of Consumer Research* 5, pp.103-122.

[36] Grönlund, J., and Rönnberg S. D.: Open Innovation and the Stage-Gate Process: A Revised Model for New Product Development. *California Management Review* 52(3).

[37] Goodpasture, J.C., 2010. Project Management the Agile Way: Making it Work in the Enterprise. J. Ross Publishing.

[38] Haines, S., 2009. The Product Manager's Desk Reference, McGraw-Hill.

[39] Hansen, M.T., and Birkinshaw, J., 2007. The Innovation Value Chain, *Harvard Business Review*, United States, Boston.

[40] Hansen, P., and Ombler, F., 2009. A new method for scoring multi-attribute value models using pairwise rankings of alternatives, *Journal of Multi-Criteria Decision Analysis* 15, pp.87-107.

[41] Highsmith, J.A., 2000. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, Dorset House, New York.

[42] Hines, A., 2006. Strategic foresight: the state of the art. *The Futurist* 40(5), pp.18–21, 2006.

[43] Horvath, P., 1993. Target Costing. Marktorientierte Zielkosten in der deutschen Praxis. Schäffer-Poeschel Verlag.

[44] Howell, J.M., Shea, C.M. and Higgins, C.A., 2005. Champions of Product Innovations: Defining, Developing, and Validating a Measure of Champion Behavior. *Journal of Business Venturing* 20, pp.641-661.

[45] Huang, K.L., and Liao, C.J., 2008. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Comput. Oper. Res.* 35(4), pp.1030-1046.

[46] Iansiti, M., 1995. Shooting the Rapids: Managing Product Development in Turbulent Environments. *California Management Review* 38(1).

[47] Kano, N., 1984. Attractive Quality and Must-be Quality. *Journal of the Japanese Society for Quality Control*, 4, pp.39-48.

[48] Kaplan, R.S., and Norton, D.P., 1992. The Balanced Scorecard - Measures That Drive Performance, *Harvard Business Review*, United States, Boston.

[49] Keinz, P., and Prügl, R., 2009. A User Community-Based Approach to Leveraging Technological Competences: An Exploratory Case Study of a Technology Start-Up from MIT. *Creativity and Innovation Management*, 19, pp.269–289.

[50] Ladas, C., 2008. Scrumban: Essays on Kanban Systems for Lean Software Development, Modus Cooperandi Press.

[51] Larman, C., and Vodde B., 2009. Scaling Lean & Agile Development. Addison-Wesley.

[52] Leifer, R., McDermott C., Colarelli O'Connor G., Peters L., Rice, M.P., and Veryzer R. W., 2000: Radical Innovation: How Mature Companies Can Outsmart Upstarts, Harvard Business School Press.

[53] Leonard, D., Rayport, J.F., 1997. Spark Innovation through Empathic Design, *Harvard Business Review*, Nov-Dec 1997.

[54] Levinthal, D. A., and March, J. G., 1993. The myopia of learning. *Strategic Management Journal* 14, pp.95-95.

[55] Li, J.Q., Pan, Q.K., and Liang, Y.C., 2010. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Comput. Ind. Eng.* 59(4), pp.647-662.

[56] Liu, H., Abraham, A., and Wang, Z., 2009. A Multi-swarm Approach to Multi-objective Flexible Job-shop Scheduling Problems. *Fundam. Inf.* 95(4), pp.465-489.

[57] Lo, K. and Lan, Y., 2010. An approach to the R&D value based upon real option method. *Quality and Quantity* 44(3), pp. 509-520.

[58] Locher, D. A., 2008. Value Stream Mapping for Lean Development: A how-To Guide for Streamlining Time to Market. CRC Press.

[59] March, J. G., 1991. Exploration and exploitation in organizational learning. *Organization Science*, 2.

[60] MacCormack, A., Verganti, R. and Iansiti, M., 2001. Developing products on "Internet time": The anatomy of a flexible development process. *Management Science* 47(1), pp. 133-150.

[61] MacCormack, A., and Verganti, R., 2003. Managing the Sources of Uncertainty: Matching Process and Context in Software Development. *Journal of Product Innovation Management*, 20, pp.217–232.

[62] Markides, C., 2006. Disruptive innovation: In need of better theory. *Journal of Product Innovation Management*, 23, pp.19-25.

[63] Mills, H., Dyer, M., and Linger, R., 1987. Cleanroom Software Engineering. *IEEE Software* 4(5), pp.19–25.

[64] Palmer, S.R., and Felsing, J.M., 2002. A Practical Guide to Feature-Driven Development. Prentice Hall.

[65] Parry, M. E., Song, M., De Weerd-Nederhof, P. C., and Visscher, K., 2009. The Impact of NPD Strategy, Product Strategy, and NPD Processes on Perceived Cycle Time. *Journal of Product Innovation Management*, 26, pp. 627–639.

[66] Paulk, M.C., Weber, C.V., Curtis, B., and Chrissis, M.B., 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison Wesley.

[67] Pfeffer, J., and Salancik, G.R., 1978: The external control of organizations: a resource dependence perspective. Harper & Row, New York.

[68] Pichler, R., 2010. Agile Product Management with Scrum. Addison-Wesley.

[69] Poppendieck, M., and Poppendieck, T., 2003. Lean Software Development: An Agile Toolkit. Addison-Wesley.

[70] Reid, S. E., and De Brentani, U., 2004. The Fuzzy Front End of New Product Development for Discontinuous Innovations: A Theoretical Model. *Journal of Product Innovation Management*, 21, pp.170–184.

[71] Reid, S. E., and De Brentani, U., 2010. Market Visioning Competence and Market Vision: Impacts on Early Performance for Radically-New, High Tech Products. *Journal of Product Innovation Management*, 27(4), pp.500-518.

[72] Rogers, E. M., 2003. Diffusion of innovations, 5th ed., Free Press, New York.

[73] Rohrbeck, R., and Gemünden, H.G., 2001. Corporate foresight: Its three roles in enhancing the innovation capacity of a firm. *Technological Forecasting and Social Change,* 78(2), pp. 231-243.

[74] Saaty, T. L., 2006. Fundamentals of Decision Making and Priority Theory. RWS Publications, Pittsburgh, Pennsylvania.

[75] Saaty, T. L., 2008. Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World. RWS Publications, Pittsburgh, Pennsylvania.

[76] Saaty, T. L., 2010. Principia Mathematica Decernendi: Mathematical Principles of Decision Making. RWS Publications, Pittsburgh, Pennsylvania.

[77] Saren, M., 1984. A classification of review models of the intra-firm innovation process. *R&D Management*, 14(1), pp.11-14.

[78] Schon, D.A., 1963. Champions for Radical New Inventions. *Harvard Business Review* 41(2), pp.77-86.

[79] Schumpeter, J. A., 1934. The Theory of Economic Development, Harvard University Press, Cambridge, MA.

[80] Schwaber, K., and Beedle, M., 2001. Agile Software Development with Scrum. Prentice Hall.

[81] Smith, P.G., and Reinertsen, D.G., 1992. Shortening the Product Development Cycle, *Research and Technology Management* 35(3), pp.44-49.

[82] Smith, P.G., and Reinertsen, D.G., 1997. Developing Products in Half the Time: New Rules, New Tools, 2nd ed., John Wiley & Sons.

[83] Smith, P. G., 2007. Flexible Product Development: Building Agility for Changing Markets. John Wiley & Sons.

[84] Shalloway A., Beaver G., and Trott J.R., 2009. Lean-Agile Software Development: Achieving Enterprise Agility. Addison-Wesley.

[85] Siemens AG: Pictures of the Future – Strategic planning for the future at Siemens [online] Available at http://www.siemens.com/innovation/en/strategie/method.htm, [Accessed 18 June 2011].

[86] Solak, S., J., Clarke, J.-P.B., Johnson, E.L., and Barnes, E.R., 2010. Optimization of R&D project portfolios under endogenous uncertainty. *European Journal of Operational Research* 207(1), pp. 420-434.

[87] Sood, A., and Tellis, G.J., 2011. Demystifying Disruption: A New Model for Understanding and Predicting Disruptive Technologies. *Marketing Science* 30(2), pp.339-354.

[88] Thomke, S., and Reinertsen, D., 1998. Agile Product Development: Managing Development Flexibility in Uncertain Environments, *California Management Review* 41(1), pp.8-30.

[89] Tidd, J., and Bessant, J., 2009. Managing innovation: Integrating Technological, Market and Organizational Change, 4th ed., John Wiley & Sons.

[90] Trott, P., 2008. Innovation Management and New Product Development, 4th ed., Prentice Hall.

[91] Tushman, M. L., and Anderson, P., 1986. Technological discontinuities and organizational environments, *Admin. Sci. Quart.,* 31, pp.604-633.

[92] Utterback, J. M., 1994. Radical innovation and corporate regeneration. *Res. Tech. Management* 37, pp.10-18.

[93] Utterback, J. M., and Acee, H. J., 2005. Disruptive technologies: An expanded view. *Internat. J. Innovation Management*, 9, pp.1-18.

[94] Vecchiato, R., and Roveda, C., 2010. Foresight in corporate organisations, *Technology analysis and strategic management* 22:1, pp. 99-112.

[95] Vela, C.R., Varela, R., and Gonzalez, M.A., 2010. Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. *Journal of Heuristics* 16(2), pp. 139-165.

[96] Watts, S. H., 2005. TSP: Leading a Development Team, Addison-Wesley Professional.

[97] Weissenberger-Eibl, M. and Teufel, B., 2011. Organizational politics in new product development project selection, *European Journal of Innovation Management* 14(1), pp. 51-73.

[98] Westerman, G., McFarlan F.W., and Iansiti, M., 2006. Organization Design and Effectiveness over the Innovation Life Cycle. *Organization Science* 17(2), pp. 230-238.

[99] Wheelwright, S. C., and Clark, K. B., 1992. Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality. The Free Press, Simon & Schuster Inc.

[100] Witte, E., 1977. Power and Innovation: A Two-Center Theory. *International Studies of Management and Organization,* 7, pp.47 –70.

[101] Xing, L.N., Chen Y.W., and Yang K.W., 2009. Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Appl. Soft Comput.* 9(1), pp.362-376.

[102] Yu, A. S. O., Figueiredo, P. S., and De Souza Nascimento, P. T., 2010. Development Resource Planning: Complexity of Product Development and the Capacity to Launch New Products. *Journal of Product Innovation Management* 27, pp.253–266.

[103] Zhang, R., and Wu, C., 2011. A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardinessobjective. *Comput. Oper. Res.* 38(5) pp. 854-867.