# Key Decisions in Strategic New Product Development for Small Software Product Businesses

Jarno Vähäniitty Software Business and Engineering Institute Helsinki University of Technology Jarno.Vahaniitty@soberit.hut.fi

# Abstract

Small companies in the software product business risk rework and market failure due to shortcomings in integrating a strategic perspective into management of product development. Because little guidance exists for helping such companies link business strategy with product development, we propose that improvement should begin with helping the key persons maintain the 'big picture' in their everyday bustle of multiple, sometimes even contradictory roles and responsibilities. We present a framework of key decisions to address in order to reconcile the perspectives of business strategy and product development, based on strategic management, new product development and software engineering literature. The framework was tested and refined as part of a study aiming to improve the NPD management practices in three companies. Our findings indicate that using the framework helped raise the companies' awareness of the importance of long-term planning in product development and yielded relevant improvement suggestions regarding daily product development activities.

# 1. Introduction

Success in the software product business demands more than just developing products that are of high technical quality - building products at the right time with the right features is at least as important. Shortened cycle times often make effective planning and execution of new product development the difference between survival and ruin. Still, software companies are tempted to shortcut the development process and bring products to market more quickly by omitting the planning stages entirely [31]. To contrast this, strategic new product development refers to taking a deliberate and systematic approach to formulating and enacting product strategies [17]. Product strategy links the company's business strategy to product development by providing a vision and competitive positioning [30] for a specific product/service proposition and guidelines for organising the development efforts [12,36]. To position this work and set up the context, *business strat-egy* focuses on how to compete in a particular industry or product/market segment, while *corporate strategy* refers to a company's overall direction in terms of its various businesses, resources and general attitude towards growth [36,41]. By software product business we mean that the amount of customer-specific development effort is (or is intended to be) relatively small [18]. In compliance with the European Union standard, 'small' companies are considered those having less than 50 people.

Existing research [4,41] as well as our own experience suggest that small companies find integrating a strategic perspective into product development decisionmaking more challenging than larger firms with more established strategy processes. Important product development decisions are often made based on the opinions of key personnel rather than through deliberate or explicit planning, and consequently, small companies are vulnerable to extensive rework and even market failure [6,39]. While small companies may not need or even afford having dedicated strategy personnel like larger firms, we propose that maintaining the 'big picture' is challenging for the key persons in the everyday bustle of multiple and sometimes even contradictory roles and responsibilities. Combined with unclear priorities due to lack of long-range planning, overbooking of resources becomes common while some important activities do not receive enough attention. This means that although important product development decisions do get made, they are not necessarily taken deliberately, but rather inadvertently, or even through inaction. Likewise, the key personnel may not even be aware of the gamut of important decisions they face in managing their efforts in new product development.

Despite of the difficulties involved, small technology-based companies have a crucial role in the industry because of their innovativeness, popularisation of new technologies, keeping established firms on their toes [33], job creation and growth [41]. For the individual company there often exists only one shot at success [31], and statistics suggest that up to one half of businesses founded in any one year are not in business five years later [41]. Because the underlying problem appears to be an overall lack of strategic management [41], it is important to *find means for supporting strategic product development in such companies*, a task we have undertaken in this paper.

Section 2 discusses the results of a literature review conducted to collect existing support for strategic new product development in small software product businesses. Section 3 presents a framework identifying what should be paid attention to in managing new product development in small software product businesses, and its theoretical background. Section 4 discusses how the framework was applied in three case companies in order to test and refine it, as well as the results from this work. Section 5 concludes the paper by outlining its contribution and presenting directions for further research.

# 2. Existing support for strategic new product development in small software product businesses

We conducted a review of strategic management, new product development management and software engineering literature to collect existing support for strategic new product development in small companies in the software product business. The areas of literature were chosen because they all address the management of new product development from different perspectives. Our conclusions were as follows:

- Most of current management theory cannot be applied directly in smaller companies because it is founded upon a large company context [20,40]. Still, even at the beginning of the life-cycle of a company, planning should incorporate analytical elements and become more formal and sophisticated as the company grows and matures [4].
- A large number of techniques, tools and methods for aligning new product development efforts with strategy exist [11]. However, these have been designed from the perspective of large companies with multiple business units, each with possibly several product lines. Literature does not provide insight into their applicability in small companies [2].
- Much of the literature on software engineering is written from the viewpoint of large organisations and companies doing *individual projects* for specific customers [5,8,15,23,27]. Also, software engineering literature prefers the engineering point of view and generally leaves the link to business management for others to handle [37].

No context-specific guidance was found for helping small software businesses to combine the perspectives of business and product development. Techniques for aligning new product development with business strategy are not directly usable in the small software product company context because of the difference in perspective; in a small software product company, the emphasis of decision-making would most likely focus more on the contents of a single product line or even a product over several releases, rather than on deciding on a multitude of development projects concerning different product lines and their extensions [21].

Instead of tailoring techniques from literature to support strategic management of new product development in small software product businesses, we think that it is more useful to increase the awareness of the underlying important decisions that are involved. This is because adapting existing techniques requires a contextspecific understanding of the underlying issues, and applying different techniques is likely to require a caseby-case justification – in small companies doing product development, improvement efforts must be perceived immediately useful if they are to receive attention [27,32].

# **3.** Key decisions in strategic new product development

In this section, we present a framework identifying what should be decided – the *key decisions* that the literature and our own experience in working with several small software companies deem important for strategic new product development.

# 3.1. Identification of the decisions

While the *how-to-succeed* in developing products varies over time both within and across companies and industries, what issues are being decided on remains fairly constant at a certain level of abstraction [26]. We found it useful to view product strategy as the result of making important decisions in managing new product development. This is because identifying such decisions in the small software product business context would provide managers with a 'checklist' of what they should be paying attention to. The starting point for our work in compiling a framework of key product development decisions for small software product businesses was a list of generic new product development decisions, or decision areas by Krishnan and Ulrich [26]. These areas are product strategy and planning, product development organisation and project management (decisions in setting up a development project), and *concept development*, supply chain design, product design, performance testing and validation and production ramp-up and testing (decisions within a project).

From the perspective of supporting strategic new product development in small software product companies, the list contains issues of unclear relevance, it lacks industry-specific detail, and gives an incomplete picture of what kinds of decisions can have profound implications for a small organisation. For example, managing the supply chain in the software business is profoundly different from the case of manufactured products and many of the management issues critical for software development, such as quality assurance are either not emphasised correctly or discussed at all. Also, the naming of the decision areas as well as the 'setting up projects' - 'within a project' categorisation imply a large company context, possibly involving manufacturing as well. Despite of these contextual shortcomings, Krishnan and Ulrich's model, as well as the *decision perspective* it provides served as a comprehensive starting point for our work.

To overcome the difficulties, we tailored the original decision areas to comply with characteristics of small software companies in the product business as identified in literature (see for example [10,16,38,40]) and our previous experience. The validity of our initial 'key decision areas' for the small software product business context was evaluated by examining whether describing real case companies is meaningful using such a structure, and whether relevant problems and challenges can be found using the decision areas as a checklist. The results of this test, our other empirical work related to the key decision areas, and the role of these in defining the set of key decision areas presented in this paper are discussed in section 4. While it can hardly be said that any of the individual key decisions would be specific to small companies, the naming, structuring and content of the framework is largely a result of the empirical context. Although the key decision areas seem suitable to software product development in general, we have not examined the scalability of the framework to larger companies and make no claims towards this.

# 3.2. Definition of the decisions

The key decisions framework helps to reconcile the perspectives of business strategy and product development by providing a list of elements that product strategies should address. The key decisions have been grouped into the decision areas of *portfolio management, organisation, development model, product management* and *quality strategy*. Although not all of the decisions de-

scribed are neither equally topical nor strategic to a company at a given time, they should be recognised as building blocks for product strategies. A company that recognises the scope of strategic new product development can *explicitly* decide not to address certain issues at a given time, as opposed to not even being aware that such issues could be important. The contents of the key decision areas are summarised in Table 1 and explained below in sections 3.2.1-3.2.5.

area	Contents	
Portfolio manage- ment (What and when?)	Deciding about product and service propositions and their release strategies (release contents, roles, types, timing) based on the key elements of the respective business models (revenue logic, marketing & sales, delivery).and t?)The constraints and requirements set by the business model and the development process on each other should be reconciliated through release strategy (see sections 3.2.1 and 5.2.).anisa- vhom, e?)Organisational design; Roles and responsibili- ties; Team staffing; Team physical arrange- ment and location; Competences; Use of out- sourcing; Development infrastructure	
Organisa- tion (By whom, and where?)		
Development model (How to pro- ceed?)	Development rhythm: Pacing & phasing; Development control: Progress tracking and control; Communication mechanisms; Relative priority to and inter- action with other development models	Concern
Product management (What, specifi- cally?)	Technology selection:         roduct       Product architecture; Employed technologies;         nanagement       Requirements engineering:         What, specifi- ally?)       Elicitation; Specification; Release planning;         Change management       Release and configuration management	
Quality strat- egy (Deliver with what empha- sis?)	Defining "good-enough" quality; Risk management: Release criteria; Release success evaluation; Test planning: Test types, Timing, Documentation, Quality metrics	s model

Table 1. Identified key decision areas

**3.2.1. Portfolio management.** This decision area consists of formulating and enacting product strategies across the portfolio of products and services offered by the company [11] based on the requirements and constraints set by the other components of the respective business models. A *business model* is a manifestation of business strategy designed for a particular product/market situation [36]. Its four main elements are *product strategy, revenue logic, marketing & sales* and *delivery,* with product strategy defining an offering and its release strategy. This definition of product strategy is consistent with the one given in section 1 because our

model implies that a release strategy sets guidelines for how the development efforts should be organised. Deciding about *release strategies* means managing the *contents*, *timing*, *roles* and *types* of future product releases across the product portfolio of the company [37]. A *release* means passing a software build and associated documentation on to one or more parties outside of development [35].

In addition to product strategy, the other elements of the business model are likely to impose constraints and requirements for the release strategy, and subsequently for the product development process as well. However, the development process may or may not be able to respond to these demands, and possible conflicts should be settled in the release strategy. Thus, discussing the other three business model elements is in order to understand the inherent complexity of the portfolio management decision area. Revenue logic refers to how a company extracts value from its operations - its mechanisms for creating revenue and profit, the basic ideas behind product pricing, and utilising other possible sources of financing [36]. Marketing & sales describes how the marketing and sales of the offering have been organised, how the products are intended to reach their markets, and identifies the respective actors and the sales process [36]. Marketing refers to decisions on market segmentation, target market(s) and presenting the products to the customers in a way that enhances their perceived value [25]. In the software product business, connecting marketing and product development through decisions such as what new features can be marketed (and how) at what stage of their construction are of crucial importance. Also, small companies must often employ indirect sales channels in addition to direct sales. Delivery refers to how the offering will reach its customes as a working solution and the actors and processes involved, for example the physical distribution, deployment, maintenance [36], tailoring or integration of the offering. Thus, the amount of customer-specific effort required stems from these decisions. The cost and benefits of offering complementary services should be balanced, and possible long-term implications should be kept in mind [32].

**3.2.2. Organisation.** While organisational boundaries may be less visible in small companies, *organisation* remains one of the most important factors in determining whether the company is capable of operating according to its intended business model [28]. *Organisational design* describes those organisational structures and associated functions that are assumed stable over time unless explicitly changed. *Defining roles and responsibilities* is especially important in small companies because of low or-

ganisational hierarchy and the emphasised role of key personnel such as senior management. Organisational change is more often realised as re-defining roles and responsibilities than changing the organisational structures themselves. Thus, acting under the 'wrong' role in a given situation may result in fluctuation in the company's business model [28]. Team staffing is the strongest mechanism in operationalising strategy - inadequate practices in resource allocation can waste resources to unimportant tasks, while simultaneuously starving the issues that deserve attention [11,26]. The physical arrangement and location of company personnel and teams affects the effectiveness of communication, and consequently, all work [14]. Because order-ofmagnitude differences can exist between team productivity [19,29], understanding *competences* and possibly investing in team collaboration and capability can have strategic implications. Outsourcing entails both considerable risks and potential payoff regardless of its exact nature, and decisions on how outsourcing should be utilised and managed can be of strategic importance. Development infrastructure deals with the selection, acquiring and usage of development tools and environments and their sharing amongst projects. In small companies development tool and infrastructure decisions can have major implications because they easily shape the development process [16] and are often coupled to the technologies employed in the product as well. Tool and infrastructure investments are also relatively costly to small companies.

3.2.3 Development model. This decision area refers to how the product development process is structured in order to realise the intended release strategy, and consists of development rhythm and development control. Development rhythm refers to the general way development efforts are structured for a certain kind of undertaking, for instance a release project, and how the rhythm of the development is supported or enforced (pacing and phasing). Pacing means creating a rhythm to the development efforts, for example by conducting the entire development as projects with clear start and end dates, and/or pacing the daily work with various practices. Phasing defines the nature, emphasis and objectives of the periods of time created through pacing. Development control refers to what measures are used to track and control progress and what communication mechanisms there are among development team members and the rest of the organisation. Also, development control guides how a development model should interact with other development models or instances of the same development model according to its relative priority (for example, changes in resourcing).

3.2.4. Product management. Product management is a term having different meanings for things to different people. Here we focus on the aspect often referred to as inbound product management [10], consisting of technology selection, requirements engineering and release management. Technology selection consists of making technological and architectural decisions, and in small companies it is crucial that key personnel outside product development should get involved. The product development personnel should support this by taking a consulting role. Technological and architectural decisions should be based on the business requirements for the product and the needs for incrementally developing the product or extending the product line. Also, most non-functional requirements directly affect the set of choices available and should be made explicit to the extent possible. Requirements engineering, in other words deciding about the principles and processes used in eliciting product and feature ideas and prioritising them to transform the viable ones as part of the product offering, is essential to strategic new product development. This area consists of collecting potential requirements (elicitation), properly documenting them (specification), prioritising and allocating to future product releases (release planning), and keeping this allocation fit in the pressures of changing requirements, defects found, schedule constraints and the changes' implications for the whole product (change management). Release and configuration management refers to the technical process and the services, tools and methodologies used in releasing versions of the product [3], as well as the basic principles and processes applied in configuration management. The latter includes keeping track of the configurations of the product(s) on the market.

**3.2.5. Quality strategy.** Defining and operationalising "good-enough" quality is one of the major strategic decisions a software product company has to make on a continuous basis [18]. The somewhat controversial "good-enough" implies an understanding of both the relative priorities of the release in terms of quality attributes (for example, the ISO 9126 standard on software product quality) and the risks involved. Whilst most companies see *risk management* as a key issue, risk is typically treated tactically and most often in an ad-hoc or piece-meal manner [9]. Existing research on software risk management is more concerned with the tactical perspective of managing risk within a development project [34] or the methods used [24] instead of the perspective of a com-

pany making product releases. However, two basic ways to incorporate elements of risk management into quality decision-making on the strategic level are determining release criteria for the product, and measuring the success of past product releases. Understanding the strengths and weaknesses of past releases in terms of added value from functionality and (absence of) defects provides depth to release criteria and quality metrics [13]. Quality metrics refer to quantitative information characterising the state of the object being tested. This information can be used, e.g., for progress tracking and as the basis for release criteria. Test planning means specifying which types of testing are to be utilised and how, the level of documentation and reporting necessary, and the quality metrics used. Management should provide input on the consequences of different kinds of failures in product quality and oversee that these are evenly matched with different types of testing and their usage. The amount and type of documentation utilised and produced by testing should vary depending on the context. In some cases, complete test case specifications and comprehensive reporting are necessary, while in others, exploratory testing [22] can be more appropriate.

# 4. Application experiences

To refine our initial framework as well as test it for completeness, we conducted case studies in three small software product businesses and arranged an experience exchange workshop. These efforts and their results are described in this section.

# 4.1. Approach and changes to the framework

After an initial outline of the key decision areas was compiled based on Krishnan and Ulrich's new product development decisions and characteristics of small software product businesses, semi-structured interviews were conducted in three software product companies in mass-market type business to understand how they handled the respective decision areas and the related software engineering practices. The case companies were chosen because of their accessibility as industrial participants to our research project, and they were in massmarket type business. In all three of the companies, called here Slipstream, Cielago and Cheops (having 20, 10 and 30 developers at the time of the interviews, respectively), the head of product development was interviewed, and at one of the companies this was complemented with interviewing one project manager and two product managers because of the extent of involvement in our research project. The data from the interviews

was thematically coded using the elements of our initial key decision areas, and where necessary, decision areas (or parts thereof) were added, renamed and restructured to better cover the issues of importance that surfaced during the interviews.

Case descriptions structured according to the decision areas illustrating the companies' practices and highlighting their strengths, weaknesses, problems and challenges were written, checked for correctness and presented along with improvement suggestions. The general findings were presented to all of the companies in a joint session. After this, company-specific observations and improvement suggestions were discussed individually at the companies in three-hour sessions, and soon thereafter, informal discussion-like interviews were conducted with the interviewed persons to evaluate how useful they perceived the analysis and the suggestions made. After an observation period of four to six months (depending on the case in question), a second round of interviews was conducted to find out what action, if any, had resulted from the first interviews and their dissemination. The heads of product development were interviewed, and at Cheops, one product manager, a project manager, a developer and the head of quality assurance were interviewed as well. From the perspective of the framework, the case studies resulted in adding one-fourth of the total content present in current version, and led to a make-over of the structure of the decision areas as well as much of the naming used.

Four months after the final interviews, we arranged an experience exchange workshop in which six small software companies (including one of the case companies) presented their current product development practices, and their self-perceived strengths and challenges using the structure of the framework. From the perspective of the framework, the workshop resulted in some structural changes and terminology changes to simplify the framework and make it more intuitive.

#### 4.2. Results in the case companies

This section briefly discusses the results of the case studies from the perspective of the participating companies.

**4.2.1. Improvement suggestions and feedback received.** The suggestions made at Slipstream address the use of requirements documentation, development rhythm, progress tracking, and quality strategy. The company agreed on the importance of the suggestions, and improvements were already being made. One month after the suggestions were presented, the interviewed senior product manager reported of initiatives towards managing

the product portfolio through business case thinking, starting systematic defect tracking and improving the requirements process.

At Cielago, suggestions addressed communication between sales and R&D, release planning, development rhythm, progress tracking and quality strategy. According to the company, the interviews had helped them identify weaknesses in their current way of working. The product had been developed in a technology-push manner and the R&D-sales relationship was not working well. This was to be improved, and during the upcoming months, the entire from-idea-to-after-sales process was to be defined complete with phases and associated inputs, outputs, roles and responsibilities. The most important target for improvement efforts during the first month had been how requirements for new products were handled.

The suggestions made to Cheops concerned requirements engineering and its connection to the development model. Although the suggestions were received well, the head of R&D did not initially consider their value significant because he considered himself already aware of these issues.

**4.2.2. Changes during the observation period.** The most significant changes at the companies from the perspective of the decision areas are summarised in Table 2. At Slipstream, the requirements process and release planning were made more systematic and project progress tracking was strenghtened. At Cielago, the roles and responsibilities of some of the key personnel were altered to stimulate interaction between R&D, sales and the customers, phases were introduced to the product development process and requirements for new products were specified and analysed more rigorously. At Cheops, the most significant improvements were in requirements prioritisation, organising quality assurance and project planning. Also, part of the development work was outsourced.

In addition to the changes listed in Table 2, the companies had worked towards solving minor problems and challenges that had been identified together during the interviews. Most of these had been successfully dealt with.

 Table 2. Changes during the observation period

Slipstream Cielago Cheops	
---------------------------	--

Portfolio management	Release planning by business case think- ing and product roadmapping	Changes in busi- ness models: Productisation of services; changes to sales channels	-
Organisation	-	Role changes in the customer- R&D interface	Some develop- ment work out- sourced; dedi- cated testers
Development model	Phased development process with mile- stone reviews (all products)	Definition of a six-phase idea-to- delivery process; clear handoffs to avoid rework (all products)	An Extrame programming - style planning game to priori- tise require- ments (all prod- ucts)
Product management	Decision to take a "Non-software" workaround to com- patibility problems From 'per release' - specs to a cumula- tive requirements backlog; a defined prioritisation proc- ess with the perspec- tive of sales in- cluded	Optimising sub- systems had stopped Written product plans and a review process	Strategic di- mension to feature pri- oritisation; roles for representing customers
Quality strategy	Systematic defect tracking, written final reports for spreading best prac- tices	Improved aware- ness of product quality and its implications	Testing spread more evenly over the project life cycle

**4.2.3. Benefits to the companies.** At Slipstream, the initially discussed improvements did take place but we were unable to trace cause-and-effect back to our study because of personnel changes. At Cielago, conducting interviews by going through the key decision areas was openly considered very useful by the management, and during the follow-up period, the company had successfully tackled almost all of the problems and challenges identified. At Cheops, the interviewed R&D manager told that he was both surprised and delighted at the amount of improvements traceable back to participating in the study.

Based on the feedback given by the companies in the follow-up interviews, using the key decisions framework to structure the evaluations of the companies' practices helped raise the participants' awareness of important decisions in managing new product development and resulted in relevant and actionable improvement suggestions. The key decision areas were also found useful for getting an overview of how a small software product company operates for process improvement purposes. Because the evaluation yielded suggestions and changes to both product strategy and engineering processes, we believe the framework is also suitable for identifying areas for improvement in the software process.

# 5. Conclusion

This section concludes the paper by outlining the contribution and presenting directions for further re-search.

# 5.1. Contribution

This paper adds understanding to strategic decisionmaking in the context of small software product businesses. The framework both illustrates the scope of issues involved in strategic new product development for small software product businesses and serves as a checklist for managing, evaluating and improving management of product development as well as the development processes themselves. Despite the fact that today's turbulent market and technological environment make planning hard and there is a trend towards flexible, agile development [1], we think that explicit planning at product strategy level has distinct value. Even in situations where planning is impossible, the planning process in itself can be valuable since it forces the company to identify and evaluate various options [7]. Thus, we see our framework as complementing modern development approaches, not as contradictory or as a replacement for them.

# 5.2. Future work

In addition to applying the framework for integrating a strategic perspective into product development, we are currently examining if the viewpoint it provides on reconciling strategy and product development can be used to focus process improvement efforts. A company's product development process should support its intended way of doing business, but explaining this relationship has to our knowledge received very little attention. We propose that the business strategy of the company should be reflected in its product development processes through decision-making at the product strategy level. For example, business needs provide constraints and requirements for future releases' contents, timing and quality, and achieving these goals can be supported by proper development rhythm, requirements processes and quality strategies.

Because the elements in the decision areas are interrelated, changes in one area set constraints and requirements on how the other areas can and/or should be organised. For example, the need to release a product having a near-zero tolerance for defects sets demands on the quality strategy. This in turn places constraints on the release strategy (for example, release cycle length), the development model used, and so on. We hypothesise that in well-organised software companies the degree and causality of these interactions and dependencies can be traced to business needs. Exploring how companies' business environments and desired ways of conducting business are reflected in their new product development could shed light on what kind of development processes and practices are suitable for different business models, and how development process capability in turn affects the set of feasible business models. By understanding the key factors involved, software process improvement can focus on the essentials, increasing the value of such work especially in those small companies having relatively ad hoc practices to start with.

# 6. References

- Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J., Agile Software Development Methods: Review and Analysis. VTT Publications 478, VTT, Finland, 2002.
- [2] Artto, K., Martinsuo, M., and Aalto, T., Project Portfolio Management - strategic management through projects. Project Management Association Finland, 2001.
- [3] Bays, M. E., Software Release Methodology. Prentice-Hall, 1999.
- [4] Berry, M., "Strategic planning in small high-tech companies", *Long Range Planning*, vol. 31, no. 3, (2002), pp. 455-466.
- [5] Brodman, Judith G. and Johnson, Donna L., "What small businesses and organizations say about the CMM", In Proceedings of *ICSE-16* (1994) pp. 331-340.
- [6] Brouthers, K. D., Andriessen, F., and Nicolaes, I., "Driving blind: strategic decision making in small companies", *Long Range Planning*, vol. 31, no. 1, (1998), pp. 130-138.
- [7] Brown, N. and Eisenhardt, K. M., Competing on the Edge : Strategy as Structured Chaos. Harvard Business School Press, 2002.
- [8] Carmel, E. and Becker, S., "A process model for packaged software development", *IEEE Transactions on Engineering Management*, vol. 42, no. 1, (1995), pp. 50-61.

- [9] Clarke, C. J. and Varma, S., "Strategic Risk Management: the New Competitive Edge", *Long Range Planning*, vol. 32, no. 4, (1999), pp. 414-424.
- [10] Condon, D., Software product management managing software development from idea to product to marketing to sales. Aspatore Books, 2002.
- [11] Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J., *Portfolio Management for New Products*, 2nd ed. Perseus Books, 2001.
- [12] Cooper, R. G., Product leadership: creating and launching superior new products. Perseus Books, 1998.
- [13] Cusumano, M. A. and Selby Richard W., Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets and Manages People. The Free Press. The Free Press, 1995.
- [14] DeMarco, T. and Lister, T., *Peopleware productive projects and teams*, 2nd ed. Dorset House Publishing, 1999.
- [15] Demirörs, Elif, Demirörs, Onur, Dikenelli, Oguz, and Keskin, Billur, "Process improvement towards ISO 9001 certification in a small software organization", In Proceedings of *International Conference on Software Engineering (ICSE 1998)* (1998) pp. 435-438.
- [16] Fayad, M. E., Laitinen, M., and Ward, R. P., "Software Engineering in the Small", *Communications of the ACM*, vol. 43, no. 4, (2000), pp. 115-118.
- [17] Gill, B., Nelson, B., and Spring, S., "Seven steps to strategic new product development", in Rosenau, M. D. (ed.), *The PDMA handbook of new product development*, John Wiley & Sons, 1996, pp. 19-34.
- [18] Hoch, D., Roeding, C., Purkert, G., Lindner, S., and Müller, R., Secrets of Software Success: Management Insights from 100 Software Firms Around the World. McKinsey & Co., 2002.
- [19] Humphrey, S. W., *Winning with software*. Pearson Education, Inc., 2002.
- [20] Jennings, P. and Beaver, G., "The Performance and Competitive Advantage of Small Firms: A Management Perspective", *International Small Business Journal*, vol. 15, no. 2, (1996).
- [21] Johnson, G. and Scholes, K., *Exploring Corporate Strategy*, 3rd ed. Prentice Hall International, Hempstead, UK, 1993.

- [22] Kaner, Cem, Bach, J., and Pettichord, B., Lessons learned in software testing - a context-driven approach. Wiley Computer Publishing, 2002.
- [23] Kautz, Karlheinz, Hansen, Henrik Westergaard, and Thaysen, Kim, "Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise", In Proceedings of *ICSE2000* (2000) pp. 626-633.
- [24] Kontio, Jyrki. Software engineering risk management a method, improvement framework and empirical evaluation. PhD Tech. Helsinki University of Technology, 2001.
- [25] Kotler, P., Marketing Management Planning, Analysis, Control, and Implementation, 5 ed. Prentice Hall International, 1997.
- [26] Krishnan, V. and Ulrich, K. T., "Product Development Decisions: A Review of the Literature", *Journal of Management Science*, vol. 47, no. 1, (Jan.2001), pp. 1-21.
- [27] Laitinen, M., Fayad, M. E., and Ward, R. P., "The Problem with Scalability", *Communications of the ACM*, vol. 43, no. 9, (2000), pp. 115-118.
- [28] Linder, Jane C., and Cantrell, S., "Five business-model myths that hold companies back", *IEEE Engineering Man*agement Review, no. 3, (2002), pp. 26-31.
- [29] McBreen, P., Software craftmanship. Addison-Wesley, 2002.
- [30] McGrath, M., Anthony, A. R., and Shapiro, Setting the PACE in product development. Butterworth-Heinemann, 1996.
- [31] Mello, S., Customer-centric product definition the key to great product development. American Management Association, 2002.
- [32] Nambisan, S., "Why Service Businesses are not Product Businesses", *MIT Sloan Management Review*, (2001), pp. 72-80.
- [33] Naumanen, M., Nuorten teknologiayritysten menestystekijät. SITRA, 2002.
- [34] Ould, M. A., Managing Software Quality and Business Risk. John Wiley & Sons, 1999.
- [35] Penny, David A, "An estimation-based management framework for enhancive maintenance in commercial software products", In Proceedings of *International Conference on Software Maintenance (ICSM 2002)* (2002) pp. 122-130.

- [36] Rajala, Risto, Rossi, Matti, and Tuunainen, Virpi Kristiina, "A framework for analysing software business models", In Proceedings of 11th European Conference on Information Systems, New Paradigms in Organisations, Markets and Society (2003).
- [37] Rautiainen, K., Lassenius, C., and Sulonen, R., "4CC: A Framework for Managing Software Product Development", *Engineering Management Journal*, vol. 14, no. 2, (June2002).
- [38] Regnell, B., Beremark, P., and Eklundh, O., "A Market-Driven Requirements Engineering Process: Results from an Industrial Process Improvement Programme", *Requirements Engineering*, vol. 3, no. 2, (2000), pp. 121-129.
- [39] Smith, J. A., "Strategies for start-ups", *Long Range Planning*, vol. 31, no. 6, (1998), pp. 857-872.
- [40] Ward, R. P., Laitinen, M., and Fayad, M. E., "Management in the Small", *Communications of the ACM*, vol. 43, no. 11, (2000), pp. 113-116.
- [41] Wheelen, T. L. and Hunger, D. J., *Strategic management and business policy*, 8th ed. Pearson Education, 2002.