



IBIMA
Publishing
mobile

Journal of Software & Systems Development

Vol. 2012 (2012), Article ID 898139, 47 minipages.

DOI:10.5171/2012.898139

www.ibimapublishing.com

Copyright © 2012 Vladimír Bureš. This is an open access article distributed under the Creative Commons Attribution License unported 3.0, which permits unrestricted use, distribution, and reproduction in any medium, provided that original work is properly cited

A Process View on Product Management in Software Engineering Companies

Author

Vladimír Bureš

Faculty of Informatics and Management, University of Hradec Králové,
Czech Republic

Abstract

Software development is considered as a knowledge-intensive set of activities. Therefore, several business concepts are usually used to support it. Product management is not an exception. However, qualitative and quantitative characteristics of its application are influenced by the size of an organization, since small and medium-sized enterprises differ when compared to large companies from the process perspective. Based on the analysis of the Czech business context this paper analyses the current state of the product management in small and medium-sized software developing organisations. Results reveal that the majority of investigated software developers consider as the most significant problem the process of requirement gathering, analysis and consequent implementation. This paper presents an

alternative process model which includes selected activities and establishes responsibilities. The paper also offers a brief analysis of possibilities that can be used for software-based support of described processes.

Keywords: product management, process perspective, software engineering, small and medium-sized enterprises.

Introduction

Latest developments in both technical and research aspects of software engineering allow to be considered as one of the most agile and progressive economic sectors. As it is a case in other sectors, large software engineering enterprises became global players with financial strength and substantial influence on the overall business development. These organisations usually master particular business concepts and related processes. However, this is in contradiction to situation in small and medium-sized enterprises (SMEs), which have their own specifics ranging from organisational structures and processes to culture or available financial resources. For this reason, this paper deals with product management in small and medium sized software engineering companies in the Czech business context. The

problem is that although product management is well-known concept, it is differently interpreted and not always successfully used in practice. The aim of the paper is to identify the weakest point of product management in the aforementioned setting and to suggest possible solution. The paper is structured as follows. In the next section basics of product management are summarised. Following section describes methods used in the described study. Then acquired results together with discussion are presented. Finally, the main conclusions are stated.

Literature Review

Generally, understanding and fulfilling customer needs have been recognised as one of the principle factors for product design and development to succeed in the market place (McKay et al., 2001).

As stated by Stallinger et al. (2011), a transition to product-oriented development is closely related to challenges such as:

- The coverage of the needs and expectations of many different existing and potentially new customers with the developed products and services,
- The alignment of the products and services to specific markets or market segments,
- The handling of increased functionality, variability, and complexity of products,
- The selection of an appropriate product architecture,

- The development of product variants by reusing existing solutions,
- The enhancement of products in alignment with a product portfolio, and
- The coordination of interdependent and interacting software products or of software as part of other products, e.g. mechatronic systems.

Therefore, proactive approach is of a high importance (Jain and Ramdas, 2005) and product management as a discipline has already emerged in many industries. The invention of product management is attributed to Procter & Gamble (Kittlaus and Clough, 2009). In the past several concepts influenced the work

with products or services ranging from the production oriented approach or sale based approach to marketing or even mass marketing (Mukerjee, 2009).

Product management can be defined as planning, organising, executing and controlling of all tasks that are focused on successful design, production and marketing of products or services offered by a company (Niehaus, 2005). It is essential to bear in mind that all these activities have one common aim, which is customer satisfaction. Therefore, more general approach to product management differentiating products, services, brand names, or segments needs to be applied (Gorchels, 2003). The aforementioned statement supports general conviction that product management belongs to the area of marketing (Mukerjee, 2009). Furthermore, product management possesses several

typical attributes which are for instance mass adaptation (Graman and Bukovinsky, 2005), or focusing on design and compatibility (Srinivasan, Lilien and Rangaswamy, 2006).

Product management consists of several activities. As stated by Ebert (2007) the product manager should be responsible for product requirements, release definition, product release lifecycles, creating an effective multifunctional product introduction team and – above all – preparing and implementing the business case. However, the whole lifecycle needs to be managed to conduct product management successfully. For instance, Weerd et al. (2010) elaborate methods and techniques for product management in its maturity stage. On contrary to this, Jansen et al. (2011) describe how painful and frustrating process can be the product sunseting, regardless if it happens in

times of crisis or in an organised and planned manner. Specifically, they offer a method for a software sunsetting, which was partially inspired by IEEE standard 1074 (IEEE, 1997), a process standard for the software lifecycle, which provides a concise description of the retirement process. The Gartner Hype Cycle describes particular stages of the response to new technologies. This model can be successfully applied to software products. Gartner defines the stages as follows (Fenn, 2008):

- **Technology trigger:** A breakthrough, a public demonstration, a product launch or some other event generates significant press or industry interest.
- **Peak of inflated expectations:** During this phase of overenthusiasm and unrealistic projections, a flurry of well-

publicised activity by technology leaders results in some successes, but more failures, as the technology is pushed to its limits. The only companies making money are conference organisers and magazine publishers.

- Trough of Disillusionment: Because the technology does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
- Slope of Enlightenment: Focused experimentation and solid hard work by an increasingly diverse range of organisations lead to a true understanding of the technology's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.

- Plateau of Productivity: The real-world benefits of the technology are demonstrated and accepted. Growing numbers of organisations feel comfortable with the reduced levels of risk, and the rapid growth phase of adoption begins.

New technologies positioned on the Hype Cycle do not move at a uniform speed through the cycle. A software product manager must be able to assess this speed in his/her business planning (Kittlaus and Clough, 2009).

All aforementioned activities underline the necessity of explicit product management. Moreover, Ebert (2007) found that with increasing institutionalization of a consistent and empowered product management role, the success rate of projects in terms of schedule predictability, quality and project duration improves.

Therefore, application of product management in case of software engineering companies, regardless the size, is of high importance.

Methodology

This study is based on qualitative research which can be defined as “the non-numerical examination and interpretation of observations, for the purposes of discovering underlying meanings and patterns of relationships” (Babbie, 1992).

Therefore, an open-ended observation, and analysis, searching for patterns, and processes that explain the ‘how’ and the ‘why’ are included. With the present case, research involves the use of qualitative data acquired from primary and secondary

sources, informal discussions, management documents, and participative observation.

Table 1: Framework for Product Management

Product management				
Portfolio management		Product roadmapping	Requirements management	Release planning
Activities	Partnering & contracting	Theme identification	Requirements gathering	Requirements prioritisation
	Market trend identification			Requirements selection
	Product lifecycle management	Core asset identification	Requirements identification	Release definition
				Release validation
	Product line identification	Roadmap construction	Requirements organising	Lunch preparation
				Scope change management

The main aim of the study is to find out what is the current state of product management in small and medium sized software engineering enterprises (as defined by the European Union standards). Firstly, the series of unstructured interviews with selected product managers from two distinctive industrial sectors. While companies from the area of software engineering are subjects of this study, selected pharmaceutical organisations serve as a benchmark. This stage of the study methodologically follows the framework adopted from Kittlaus and Clough (2009) depicted in Table 1. Specifically, the operational thinking is applied to differentiate between the “as is” and “to be” states (Richmond, 1993). Secondly, the comparative analysis is applied to identify major differences. Then, the analysis and consequent synthesis based on brainstorming or visualisation tools are used to describe the current state in small and medium sized software

engineering companies. Finally, the weakest part of product management is identified and the process model together with its description is developed. Additionally, simple comparative analysis of possible approaches to computer based support of the model is conducted.

Results and Discussion

At the general level, small and medium sized software engineering enterprises in the Czech Republic do not have significant problems with implementation of new business concepts, such as mobile oriented services (Kozel and Mohelská, 2010), knowledge management (Bureš, 2006), or portal supported activities (Černá and Poulová, 2008). Product management is not an exception. However, the study reveals

that this type of industry has its own specifics when compared to others (e.g. pharmaceutical enterprises). First of all, the stage of product management, in which the functional requirements are gathered and applied, is the most frequent source of problems in the overall process. Therefore, explicit description of this stage is created and three basic process models (see Figures 1, 2, and 3) of this subprocess are identified. Moreover, responsible business units related to this process are determined, and corresponding documents (forms) are created. The special attention is paid to the fact that analysis of customer need information involves three major issues: (1) understanding of customer preferences, (2) requirement prioritization, and (3) requirement classification (Xu et al., 2009). In the following section, selected

subprocesses related to the product management department are briefly characterised.

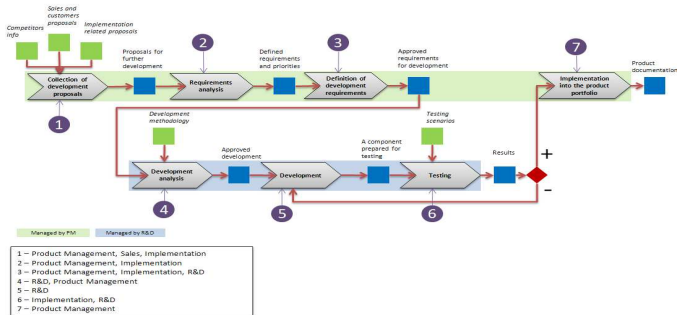


Fig 1. Process of Functional Requirements Incorporation

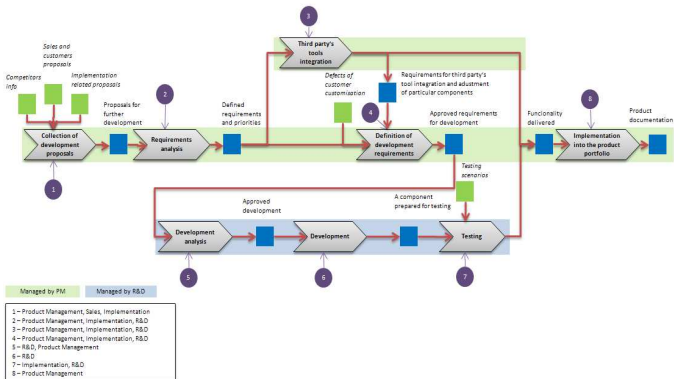


Fig 2. Process of Functional Requirements Incorporation – Integration of a Third Party's Tool

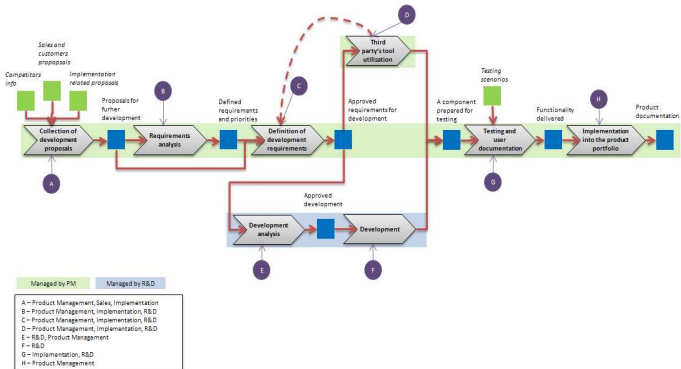


Fig 3. Process of Functional Requirements Incorporation – Utilisation of a Third Party’s Tool

Collection of Development Proposals

The main goal of this activity is to create a repository of proposals for further software development. Both proposals of new components and suggestions for modifications of existing components are welcomed. A proposal is considered as a short specification of a functionality, which is currently missing. Details do not have to be specified at this stage. However, it is necessary to avoid proposals related to bugs corrections or specific customer customisation. These should be collected in particular customer supporting systems such as Help Desk applications. As indicated in all figures, there are several sources of proposals and in fact, there should not be any limitation or restriction in terms of volume, time, working position, or department of origin.

In the specialised form every proposal is recorded under unique ID together with its additional characteristics, such as relevant component, name of the proposal, source, brief description, date, current state, related documents, etc. Particular attributes have their own scale that is used for better management of the proposal. For instance, current can be set to Activated, Refused, Approved – pending, Approved – completed.

Requirement Analysis

Product management employees evaluate assigned proposals regularly and decide, which of them can be realised, how consistent they are with an existing strategy of product development, or what are their priorities. Product managers in product software companies usually have to prioritise larger

amounts of requirements. The study reveals that this is probably the weakest point of product management in SMEs, since the way in which this is done is not systematic, or responsibilities are not explicitly defined. Although a number of agile requirements prioritisation techniques exist, which can be classified into two main categories (Racheva et al., 2008): techniques used to prioritise small amounts of requirements (small-scale) and techniques that scale up very well (medium-scale or large-scale), thus can be used for the prioritisation of larger amounts of requirements, majority of product manager are not aware of them and use simple and ad hoc methods. For instance, Bebensee et al. (2010) discuss how a technique called Binary Priority List can be applied by product managers of software companies. They investigate the research question, how can Binary Priority List be applied as a requirement prioritisation technique in small

product software companies and how reliable are its results. Eventually, they concluded that Binary Priority List is a suitable technique for prioritising medium amounts of requirements and could especially help smaller software product companies to formalise their requirements prioritisation process. Furthermore, Kano model has been widely practiced in industries as an effective tool of understanding customer preferences. Four types of product attributes are identified (Kano et al., 1984): (1) must-be attributes are expected by the customers and they lead to extreme customer dissatisfaction if they are absent or poorly satisfied, (2) one-dimensional attributes are those for which better fulfillment leads to linear increment of customer satisfaction, (3) attractive attributes are usually unexpected by the customers and can result in great satisfaction if they are

available, and (4) indifferent attributes are those that the customer is not interested in the level of their performance.

However, from the software engineering perspective the Kano classification provides limited decision support in engineering design and it fails to account for the producer's concerns in terms of the capacity to fulfil the customer needs. Therefore, Xu et al. (2009) propose an analytical Kano (A-Kano) model. The A-Kano model extends traditional Kano model by introducing:

- 1) Kano indices, which are quantitative measurements of customer satisfaction derived from Kano questionnaires and surveys;

- 2) Kano classifiers, which consist of a set of criteria to classify customer needs based on the Kano indices;
- 3) Configuration index, which provides a decision factor for selecting the functional requirements that contribute to product configurations; and
- 4) Kano evaluator, which is a shared surplus-based performance indicator leveraging upon both the customer's satisfaction and the producer's capacity.

Nevertheless, all proposals do not have to undergo this process. Occasionally, it is apparent that a proposal needs to be satisfied and therefore it can process directly to the stage Definition of development requirements.

Definition of Development Requirements

The aim of this activity is to prepare all required materials for Implementation or R&D department. If the third party's tool is supposed to use, the decision about its integration or utilisation needs to be done during this stage. Further development will vary according to current case – own development (Figure 1), integration of a third party's tool (Figure 2), or mere utilisation of a third party's tool (Figure 2). Regardless the case, the final definition is developed for the complete functionality. Moreover, functional and non-functional requirements need to be differentiated.

The main outcome of this stage is a form containing the list of requirements which are supposed to be either implemented by

the Implementation department, or developed by the R&D department. The list comprises of the requirement ID, name of the component, brief description, date, owner, or current state, which can be set to In development, In testing, Documentation creation, Processed, Implemented, Refused.

Third Party's Tools

The aim of this phase is to acquire “know how” embedded in other products or solutions. These products need to be analysed and the decision whether integrate or merely utilise it for our purposes has to be done. While utilisation is based mostly on connection of the product in as simple way as possible (e.g. development of plug-ins), the integration is more complicated process focused on modification of product functionality or user

interface. However, in both cases several issues need to be carefully considered – legal and licence issues, hardware and software requirements, financial constraints, time limitations, or default configuration specifics of the product. The main outcome of this stage is a component prepared for development, or even testing. Both activities are consequently performed by competent departments.

Implementation into the Product Portfolio

Once all selected requirements are implemented or third party's tools are used, the modified product can be included into the product portfolio. Product management department is responsible for the final stage of this particular activity. First of all, product related documentation has to be changed

appropriately, employees from implementation and sales departments need to be informed and trained, as well as existing customers have to be informed and their installations upgraded. Apparently, all these activities need to be documented and carefully monitored to ensure their successful finalisation. Software product line engineering (Clements and Northrop, 2002) which generally enables software developing organisations to gain significant improvements with respect to development and enhancement costs, time-to-market, and product quality, can be applied at this stage.

Due to verification purposes the aforementioned model has been applied in the selected small software engineering organisation – Koncept Hradec Králové. The application was focused on the overall product portfolio, however the main

attention was paid to the complex information systems provided to brokers operating on the Czech insurance market. In relation to this product particular roles and their connections to basic documents was established (see Table 2). Due to this changes the most important improvements were identified in areas such as internal communication, product controlling, responsibility setting, document quality, and increased turnover connected with the product. However, these are only preliminary results. Deeper analysis requires additional data and time. Therefore, results from the verification stage are not further elaborated in this paper.

Table 2: Roles and Documents Related to the Particular Product

Documents	Role			Frequency
	Responsible	Assistant	Awaiting	
User documentation	PG	DG	SG	Continuously
Technical documentation	DG	PG	SG	Continuously
Price list	SG	PG	TM	Once in a set period
Market analysis	SG	PG	TM	Once in a set period
Product analysis	PG	SG,DG	TM	Once in a set period
Realisation plan	TM	PG	DG,SG	Four times during the year
List of requirements	PG	SG,DG	TM	Four times during the year
Pipeline	SG		TM	Once in a set period
Initial study*	PG	DG	TM	According to the needs
Feasibility study*	DG	PG	TM	According to the needs

* in case of new products/modules only

PG – Product Guarantor; DG – Development Guarantor; SG – Sales Guarantor; TM – Top Management

Software for Product Management

It is apparent from the previous section that product management is connected with plenty of documents, forms and their versions. Therefore, the attention of this study is also aimed to computer based support of this administratively intensive issue. Although the solution of this problem seems to be obvious, since software engineering companies should be able to produce their own tools for these purposes, conducted comparison of possible approaches to document and workflow management reveals that one cannot apply this straightforward thinking. The problem is that SMEs do not possess with resources available for additional work on this task. Product management cannot overload employees from the process perspective and financial limitations have to be

considered. Therefore, several products are excluded a priori (e.g. robust and expensive solutions such as SAP Product Lifecycle Management, HP Application Lifecycle Management 12, Sybase Product lifecycle management, Oracle Project Portfolio Management, Rational Focal Point, or Accompa). Applied comparative criteria are financial burden, intuitive and easy-to-use interface, and functionality. The analysis considers following alternatives: file system (based on shared discs), own ad hoc solution, Wiki based systems (e.g. Google docs), licenced applications (MS Sharepoint), open source applications (Open PPM). Results of the analysis are summarised in the Table 3. Apparently, micro and small software engineering companies should support product management with a Wiki approach based on services provided by external organisations. This approach is not demanding in terms of costs and time and

provides users with acceptable functionality. Medium sized enterprises can prefer commercial products. Although there are additional costs related to its implementation, the functionality can cope with growing amount of documents that are used in these companies on daily basis.

Table 3: Comparison of Approaches to Product Management Computer Based Support

	Major advantages	Main disadvantages	Additional remarks
File system	Implementation speed. Simplicity at the initial stage. Low learning requirements.	Obstacles to management connected with growing number of files (e.g. version management). Usually inappropriate metadata. Low possibilities for searching inside documents.	Time needed for implementation: Hours. Licence: For free. Recommendation: Inappropriate.
Own ad hoc solution	Functionality fully respects requirements.	Time consuming. Taking away sources needed for business cases.	Time needed for implementation: Weeks. Licence: own product. Recommendation: Appropriate only if there are specific requirements which cannot be satisfied by other solutions.
Wiki based system	High implementation speed. Low learning requirements. Automatic versions of documents and their contents. Low risk of data loss (back-ups usually created by providers). Low costs.	Data stored "outside the building". Occasionally problems with searching.	Time needed for implementation: Hours. Licence: For free. Recommendation: Appropriate for micro or small companies.
MS Sharepoint	Broad functionality. Good version management. Good searching – metadata included. Possibility of connection to workflow processes. Integration with external systems.	Necessity to master administration. Complexity of the system needs time to cope with it. Connection to hardware and need to solve solution of back-ups.	Time needed for implementation: Days. Licence: commercial product. Recommendation: Appropriate for medium companies or companies with Windows Server licence.
Open PPM	Freeware. Low costs.	Project is still in the testing stage.	Time needed for implementation: Days. Licence: GPLv3, for free. Recommendation: Inappropriate due to its current instability.

Conclusions

Product management is a concept, which if appropriately handled, can help organisations to better respond to customer needs and thus increase competitiveness. In case of Czech small and medium sizes software engineering enterprises advanced business concepts such as product management or knowledge management are considered differently, and only some processes or activities are managed explicitly (Bureš and Brunet-Thornton, 2009). Nevertheless, the main problem remains mostly the same. The process of feedback shaping the further setting and functional attributes of products is based on ad hoc or non-systematic approach. Therefore, the process model elaborated with the help of selected methods was presented and explained. Additional conclusion is related to document workload connected

with product management activities. The support of document flow can be based on several approaches. The study reveals that while micro and small sized companies can use wiki based systems available on Internet, medium sized companies are encouraged to use commercial application due to their functionality and security issues.

Acknowledgment

This paper is supported by the research project 5.1 SPK01/013 “Investigation of methods, procedures and organisational structures suitable for optimisation of product management processes influencing the management of new ICT products development” coordinated by the Hradec IT cluster.

References

Babbie, E. R. (1992). *The Practice of Social Research*, Wadsworth Inc., Belmont.

Bebensee, T., Van der Weerd, I. & Brinkkemper, S. (2010). "Binary Priority List for Prioritizing Software Requirements," *Lecture Notes in Computer Science*, 6182, 67-78.

Bureš, V. (2006). 'Knowledge Management and Its Implementation,' *Proceedings of the 2nd International Conference on Web Information Systems and Technologies*, Setubal, Portugal, 115-118.

Bureš, V. & Brunet-Thornton, R. (2009). 'Knowledge Management: The Czech Situation, Possible Solutions and the Necessity for Further Research,' Proceedings of 6th International Conference on Intellectual Capital, Knowledge Management and Organisational Learning, Montreal, Canada, 95-102.

Černá, M. & Poulová, P. (2008). 'Visit Rate of Internet Portals and Utilization of Their Tools and Services,' *E + M Ekonomie a Management*, 11 (4), 132-143.

Clements, P. & Northrop, L. (2002). Software Product Lines: Practices and Patterns, *The SEI Series in Software Engineering*, Addison-Wesley, Reading.

Ebert, C. (2007). "The Impacts of Software Product Management," *Journal of Systems and Software*, 80 (6), 850-861.

Fenn, J. (2008). Understanding Gartner's Hype Cycles, Research document G00158921, *Gartner, Inc.*

Graham, G. A. & Bukovinsky, D. M. (2005). "From mass production to mass customization: Postponement of inventory differentiation," *Journal of Corporate Accounting & Finance*, 17 (1), 61-65.

Gorchels, L. (2003). 'Transitioning from Engineering to Product Management,' *Engineering Management Journal*, 15 (4), 40-47.

Jain, S. & Ramdas, K. (2005). "Up or Out - or Stay Put? Product Positioning in an Evolving Technology Environment," *Production and Operations Management*, 14 (3), 362-376.

Jansen, S., Popp, K. M. & Buxmann, P. (2011). "The Sun also Sets: Ending the Life of a Software Product," *Lecture Notes in Business Information Processing*, 80 (2), 154-167.

Kano, N., Seraku, N., Takahashi, F. & Tsuji, S. (1984). 'Attractive Quality and Must-Be Quality,' *Journal of the Japanese Society for Quality Control*, 14 (2), 39-48.

Kittlaus, H. B. & Clough, P. N. (2009). *Software Product Management and Pricing: Key Success Factors for Software Organisations*, Springer Verlag, Heidelberg.

Kozel, T. & Mohelská, H. (2010). 'Models of Firms with Mobile Oriented Architecture,' *E + M Ekonomie a Management*, 13 (4), 135-142.

McKay, A., De Pennington, A. & Baxter, J. (2001). "Requirements Management: A representation scheme for product," *Computer Aided Design*, 33 (7), 511-520.

Mukerjee, K. (2009). Product Management: Text and Cases, *PHI Learning Private Limited*, New Delhi.

Niehaus, E., Pohl, K. & Böckle, G. (2005). 'Product Management, Software Product Line Engineering: Foundations, Principles and Techniques,' Pohl, K., Böckle, G. and van der Linden, F. (eds), *Springer Verlag, Heidelberg*, 163-192.

Racheva, Z., Daneva, M. & Buglione, L. (2008). "Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products," *Proceedings of the Second International Workshop on Software Product Management 2008*, Barcelona, Spain, 49–58.

Richmond, B. (1993). "Systems Thinking: Critical Thinking Skills for the 1990s and Beyond," *System Dynamics Review*, 9 (2), 113-133.

Schultz, D. J. (1997). 'IEEE Standard for Developing Software Life Cycle Processes,' *IEEE Std 1074-1997*.

Srinivasan, R., Lilien, G. L. & Rangaswamy, A. (2006). "The Emergence of Dominant Design," *Journal of Marketing*, 70 (2), 1-17.

Stallinger, F., Neumann, R., Schossleitner, R. & Zeillinger, R. (2011). "Software Life Cycle Activities with Product Strategy and Economics: Extending ISO/IEC 12207 with Product Management Best Practices," *Communications in Computer and Information Science*, 155 (5), 157-168.

Van der Weerd, I., Bekkers, W. & Brinkkemper, S. (2010). "Developing a Maturity Matrix for Software Product Management," *Lecture Notes in Business Information Processing*, 51 (3), 76-89.

Xu, Q., Jiao, R. J., Yang, X., Helander, M., Khalid, H. M. & Opperud, A. (2009). "An Analytical Kano Model for Customer Need Analysis," *Design Studies*, 30 (1), 87-110.