Productization: Transforming from Developing Customer-Specific Software to Product Software

Peter Artz[1], Inge van de Weerd[1], Sjaak Brinkkemper[1], and Joost Fieggen[2]

[1] Department of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{pgjartz,i.vandeweerd,s.brinkkemper}@cs.uu.nl
[2] MP Objects, Rotterdam, The Netherlands
joost.fieggen@mp-objects.com

Abstract. Developing product software is getting increasing attention from both academics and practitioners. Organizations are recognizing the benefits and importance of developing a product for a market. Also, several software companies that develop customer-specific software have identified a need to change to developing and selling product software. Since few studies are available in this domain, it is difficult for organizations to handle such a transformation. In this research, we introduce the productization process that describes the transformation from developing customer-specific software to product software.

Keywords: productization, software product management, customer-specific software, product software.

1 Introduction

Custom information systems are made-to-order systems and are typically built for specific users [1]. Nowadays, inspired by the success of large software vendors, such as Microsoft and SAP, an increasing number of companies that develop this customer-specific software have a desire to transform to developing product software. *Product software* is defined as "a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market" [2]. Both situations differ in many respects [3], for example where for customer-specific software development usually one release is delivered, the product software development consist of recurrent releases. Consequently, this involves proper requirements management and release planning. Therefore, organizations that make a shift to developing product software need to change their internal processes.

An approach that can be used to structure the internal processes of a product software company is the reference framework for Software Product Management (SPM) [4], which gives product managers and organizations more insight information about the product management processes and involved stakeholders. In the reference framework, key process areas, stakeholders and their relations are modeled. The four main product management functions are Portfolio management, Product roadmapping, Requirements management, and Release planning. Additionally, the interaction

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 90–102, 2010.

of product management with internal stakeholders (company board, research & innovation, development, support, services, and sales & marketing) and external stakeholders (market, partner companies, and customers) are also included.

Currently there is barely any scientific literature available on the explicit transformation from developing customer-specific software to developing product software, which we coin as productization: *the transformation process from customer-specific software development to a standard software product* [5]. As a result, the main research question of this study is:

How can organizations transform from developing customer-specific software to product software?

In this research, we focus on the software product management domain. Evidently, there are more processes that change during a productization process, such as services and sales, but these are not in the scope of this research.

In the following section we first describe the research approach that was followed. Section 3 presents our main result: the productization process. Subsequently, in section 4, this result is evaluated. In section 5, we present the discussion. The final section consists of the conclusions and suggestions for future research.

2 Research Approach

The research described in this paper is a design science research. In order to answer our main research question, we apply the design-science research guidelines proposed by Hevner et al. to describe the problem solving process of understanding, executing, and evaluating research [6]. The design-science guidelines "seek to extend the boundaries of human and organizational capabilities by developing new innovative artifacts". The artifact that we develop in this research is the *Productization process*, which describes the transformation process that companies typically experience when shifting from developing customer-specific software to product software. We combined the five process steps of Vaishnavi and Kuechler [7] with the guidelines from Hevner et al. [6] to give our approach more structure:

1. *Awareness of the problem.* The initial step consists of all activities in order to understand the problem. The main technique we applied in this step is an extensive literature study on several research areas. As a result of that, we defined the problem definition and we determined the contribution of this study. Two guidelines which are covered within this step are 'problem relevance' and 'research contributions'.
2. *Suggestion.* The second step proposes a solution from the existing knowledge and theory base for the problem area. By carrying out a literature study, we identify the differences between developing customer-specific software and developing product software. In addition, we study various methods and approaches that can be used to change the internal processes of an organization. Consequently, this step covers the guideline: 'design as a search process'.
3. *Development.* Based on the suggestions from the previous step, we develop a first version of the artifact, which is, in this case, the productization process.. We use two data collection techniques in order to gain additional knowledge. Firstly, we carry out exploratory interviews among software product managers and software

product management experts, to gain more insight on how organizations evolve over time. Secondly, a literature study is used to obtain more information on the transformation process. The result of this step was the productization process as proposed in Section 3. This third step covers the guidelines ‘design as an artifact’ and ‘research rigor’.

4. *Evaluation.* The fourth step consists of the evaluation of the designed artifacts. We interview several SPM experts, create a survey for software product managers, and carry out a business case. The evaluation is described in Section 4. Accordingly, this step covers the guidelines ‘design evaluation’ and ‘research rigor’.
5. *Conclusion.* Finally, the last step analyzes the results of the study. We use them to deduce the answer towards our research question and provide a discussion of the results. As a result, this step covers the guideline ‘communication of the research’. The discussion and conclusion are described in sections 5 and 6.

3 Productization

Two major differences exist between customer-specific software development and product software development: the difference between stakeholders and a major pressure on time to market [1]. The main stakeholder involved for customer-specific software development is the external stakeholder. They specify which requirements need to be implemented and are more involved during the development. Consequently, the main pressures for such organizations are the costs and the customer satisfaction. On the other hand, within a product software company, the developer decides about the requirements and selects the stakeholder representatives. The main pressure for these companies is the time to market, which is covered by regularly releasing new releases. Therefore, it requires a careful release planning and requirements prioritization [3].

In order to evolve customer-specific software into product software, we propose the productization process, consisting of different stages in which the transformation process is depicted (Fig. 1). This process assumes an internal drive to change from developing customer-specific software to a standard software product. The external triggers that can lead to transformation, such as customer and market dynamics, are not taken into account.

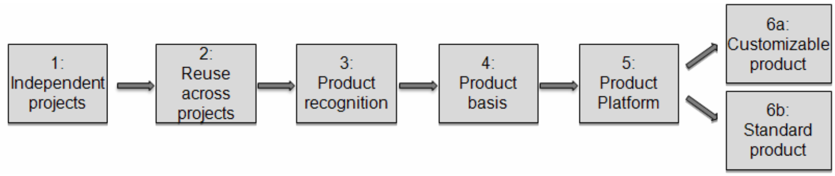

Fig. 1. Productization process

Organizations are able to enter at each stage of the process and continue until one of the two end stages is reached. The two possible end stages of the productization process are: *customizable software product* (stage 6a) and *standard software product* (stage 6b). According to Hietala et al. [8], this “degree of productization”, is influenced by product market, concepts, benefits, positioning, requirements, features, specifications, delivery channel, marketing, selling, and packaging. We applied two end stages for the productization process because often there is a need for some customization in order to integrate software in a customer-specific situation. It can take substantial time and effort to get enterprise product software up and running and therefore pieces need to be customized [9]. Also [10] recognized the intension of selling software products and of providing support services. They identify this approach as a "hybrid approach" and consist of a customer-specific part so that the product is better applicable within current infrastructures.

In the following sections we depict and describe each productization stage. We use a red color for the customer-specific parts and blue for the shared parts.

3.1 Stage 1 – Independent Projects

This first stage of the productization process represents a portfolio of projects that are executed independently. A company is carrying out projects that have barely any standard common functions or features. The software applications that are delivered differ considerably in size, budget, technology, and functionality. These differences are represented in the figure by using different sizes for the software blocks.

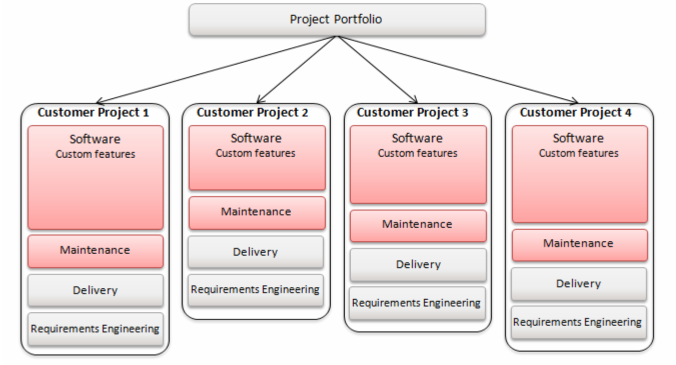

Fig. 2. Independent projects

The projects are especially driven by the customers and therefore they are the main stakeholders within this phase. The success of the projects is measured by customer satisfaction and acceptance. Consequently, there is usually a small physical distance between customers and the developers [11]. Keeping the customers involved is essential in order to satisfy them and therefore all customers' requirements must be obtained. Based on the fixed number of requirements, the delivery date and the required resources can be determined [12]. After the software is developed, it is validated together with the customer and after acceptance it is implemented.

3.2 Stage 2 – Reuse across Projects

The second stage of the productization process is identified as *Project Feature Reuse: Projects are executed differently and features or functionalities are reused across*

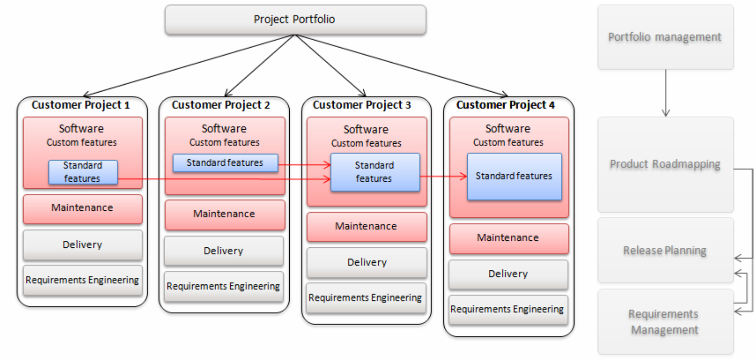

Fig. 3. Reuse across projects

projects. In contrast to stage 1, there is the possibility of reusing specific, already developed, features, functions, components or modules from previous projects. Hence, this stage represents a company that starts to develop projects by reusing elements from other projects. An advantage of reusing functionalities from finished projects is that the overall quality and reliability of the software can increase because specific parts have already been tested within previous projects. This pattern of reusing features from previous projects continues throughout the upcoming projects resulting in a basic set of standardized features or core functionalities. However, at this stage, the amount of customized features developed is still considerably larger than the standardized features.

3.3 Stage 3 – Product Recognition

Compared to stage 2, in stage 3 the standardized part of the projects is larger than the customized parts. We therefore call this stage as *Product Recognition: Large parts of projects are reused and a product scope starts to emerge from the reused functionalities*. Basically, a company is able to identify the similarities of customers’ wishes, which results into a more generic basis for / recognition of a product. Important in this stage is that an organization should evaluate the potential to become market-driven. When it is decided to start transforming to product software, stage 3 is the first step of creating a product. During this stage, the main boundaries of the product emerge and based on that a company can define a product for a specific market or purpose.

Within each of the flowing stages, the maturity of the product management functions should increase until at the end all functions are fully in place. We do not provide a specific adoption approach because there is no given hierarchy of implementing the product management functions. In the figure, we indicate this growth in maturity with the gray block on the right-hand side of the figure. The further in the productization process, the higher the product management maturity and the larger the dark segments of the different SPM blocks.

Central in stage 3 is the requirements management function. All new emerging customers’ requirements should be stored and managed in one central place. Managing the requirements is necessary because when a company starts to recognize a

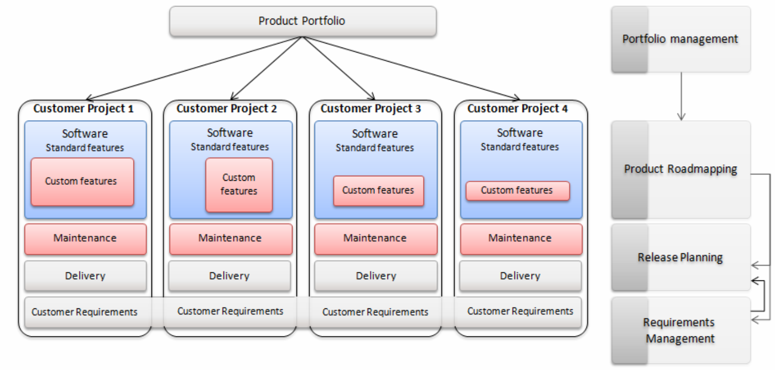

Fig. 4. Product recognition

specific product there is still a need to satisfy the customers. The main focus of this stage is still based on satisfying the customers. Therefore, we cannot speak of a standardized product designed for an entire market. Each project still has its own customized part merged into the main structure of the software in order to satisfy the customer.

3.4 Stage 4 – Product Platform

One of the major differences between developing customer-specific software and product software is the change in the lifecycle of a product. In order to determine the future of a product, a company should start focusing on future releases in order to gain bigger market share. Based on the definition of [13], we define this stage as *Product Platform*: *a set of features that form a common structure, from which a stream of derivative products can be efficiently customized, developed and produced.* As in the previous stage, many features are reused across projects. What is more, the product is recognized and a clearly defined product platform is used as a basis.

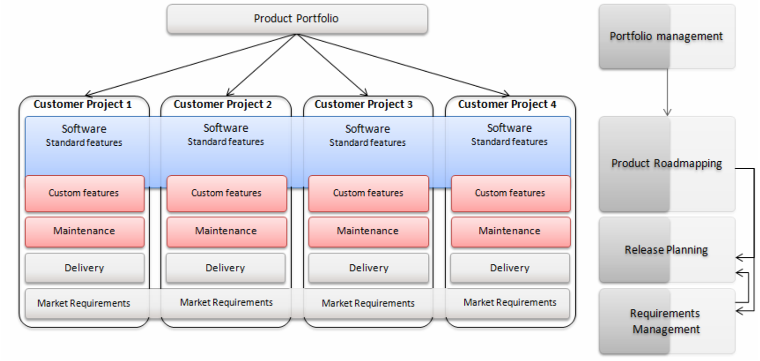

Fig. 5. Product platform

Due to the introduction of the roadmapping function, a company can generate a long-term plan. This means that the main focus on satisfying the customer is decreasing and the focus of gaining more market share should increase. The requirements management functions should also gather market requirements instead of only the customer requirements. These market requirements are needed in order to determine the content of the future software product releases because eventually the customized part should be as small as possible.

At this stage, we still are not able to speak of a first release of a software product; the amount of customized features within the projects is at this point still too large. The standardized part of this stage can be seen as the initial start of creating the product by generating its main core. This core can also be identified as a generic product platform which is extended by a large customer-specific layer.

3.5 Stage 5 – Standardizing Product Platform

The main focus of this stage is still the customer-orientation, but compared to stage 4, it starts to change towards a market orientation and bringing the emerging product to the market. Therefore, the company should create a more standardized product which is applicable for more customers in the market. The definition of this stage originates from the definition of [13]. We define it as *Standardizing Product Platform: increasing the set of features that form a common structure and introduce releases, from which still a stream of derivative products can be efficiently customized, developed and produced.*

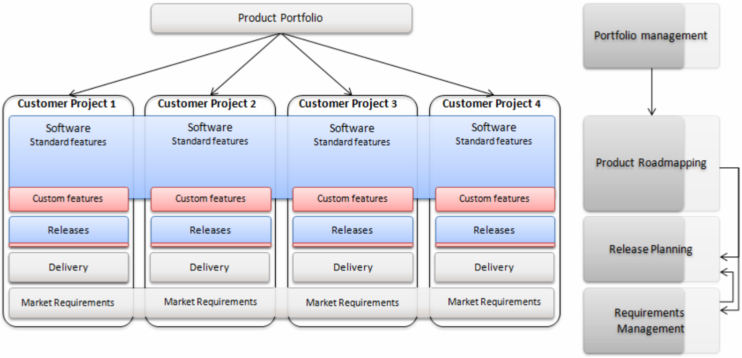

Fig. 6. Standardizing product platform

If necessary, the organization can still create for each customer its own customized part. The result of this approach is a standardized main product with additional event-based (customizable) releases. Due to the introduction of the releases, the lifecycle of the product increases and current customers get acquainted with these releases.

3.6 Stage 6a – Customizable Software Product

Stage 6a is one of the end stages of the productization process. By adapting the definition of [2], we define this stage as *Customizable Software Product*: *a packaged* *configuration of a software-based service, with auxiliary materials, which is released for and traded in a specific market and customized for a specific customer.* In contrast to the previous stage, in stage 6, releases are the same for each customer.

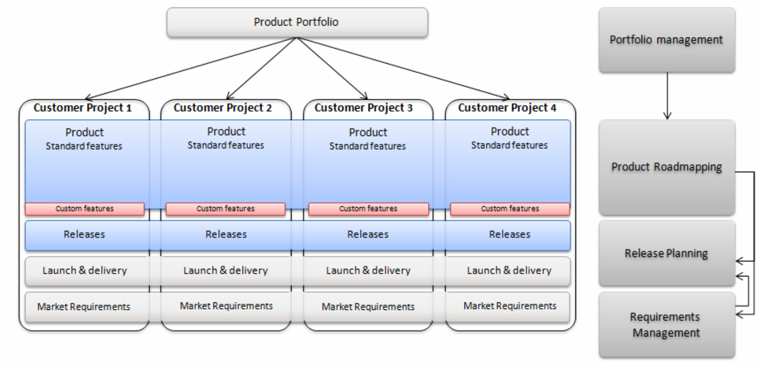

Fig. 7. Customizable software product

For some companies, there is still a need to be able to customize the software product so that it can be integrated within specific situations. This product type is characterized by software that is too complex to be sold ‘off-the-shelf’ and that requires customization or special integration and installation work [10]. Therefore, there is the need for a customized layer on top of the product so that the required functionalities can be added. The size of the functionality and code of is customization is, in general, small in comparison to the standard product.

Also [14] identified the essence of still having a customizable part in order to be able to apply the product to different situations. Additionally, [9] identified this type of product software as “enterprise solution systems” because usually this type of product software is developed for other enterprises.

3.7 Stage 6b – Standard Software Product

As described in Section 3, the productization process has two end stages. Stage 6b is the other end stage of the productization process. Inspired by [2], we define this stage as *Standard Software Product: a packaged configuration of software components, which is released for and traded in a specific market.*

After several stages still focusing on the customers, this stage focuses on a market in general. By performing active marketing and sales activities, the company should start to sell the product to a mass-market and start looking at the wishes of that market. In order to bring the product to the market, it is required that there are no customized features included within the product and the product is completely configurable. Furthermore, the main measurement in order to determine success is mainly based on getting a bigger market share and having a shorter time-to-market. An advantage of selling product software is that no additional development is required in order to be able to sell the product to new customers. This results in a much bigger potential market and more customers [8].

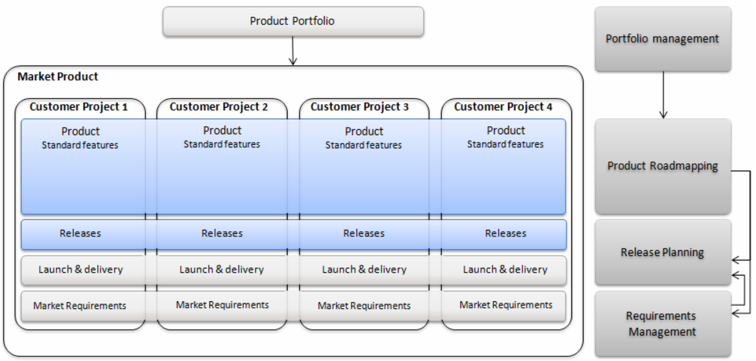

Fig. 8. Standard software product

4 Evaluation

The methods we used for the evaluation of the artifacts are expert validation (consisting of interviews and a survey), and a business case. In this section, we describe both types of validation and the related validity threats.

4.1 Expert Validation

For our expert validation of the productization process, we used an expert panel of five experts, because the interviewed experts provide an informed, objective, and unbiased opinion as well as suggestions about the developed artifact [16]. The focus of the interviews was to evaluate the acceptance and recognition of the productization process in the past of organizations. Key questions were related to how Dutch companies actually transformed in order to become a product software company. Additionally, we evaluated whether the productization process is technically sound in relation with the viewpoints of the SPM experts.

Secondly, we created a survey in order to get the opinions of eight product managers from a wide range of companies which participated on an SPM course. The survey was designed in such a way, that we were able to examine the structure of artifact for static qualities (e.g. complexity or readability) [6]. We wanted to find out whether the participants were able to recognize such process and the possibility to select a stage which is applicable for their product(s). As a result, several suggestions emerged and all participants were able to select an applicable stage. More details on the expert validation can be found in [5].

4.2 Productization in Practice: A Case Study

In addition to the expert validation, we also carried out a case study at MP Objects to validate the applicability of the productization process in a business environment. By performing a case study, the validation process provides an in depth evaluation in a business environment [17]. MP Objects is a small sized software vendor which is a specialist in providing ICT solutions for the logistic sector. The software is a unique

IT solution for Supply Chain Management within and across organizations and they support companies with the integration across the supply chain, the collaboration between people in the order cycle, and the optimization of processes in the supply chain.

In order to apply the productization process, we designed a specific approach that consists of three steps. The first step determines the initial position by carrying out an assessment. The assessment uses the maturity matrix for determining the maturity of the (present) SPM processes [19], situational factors for the determination of the best suitable maturity levels [18], and the process deliverable diagram for the visualization of the organizational structure of the initial situation [20]. Secondly, a gap analysis must identify the distance from the initial position until being fully Software Product Management oriented. We used situational factors [18] to determine the best suitable maturity levels for the product management functions. Consequently, with the initial maturity levels we determine which processes need to be implemented or improved. Finally, based on the gap analysis, recommendations can be identified, which an organization can use when they continue in transforming to become market-driven. An elaborate case study description can be found in [5].

4.3 Validity Threats

In conducting our case study, we considered the three validity threats as described by [21]. Firstly, *construct validity* is covered by using multiple sources of evidence for the data collection of specific information. While gathering the required data for the productization approach, we applied the viewpoints of consultants and developers in order to get a better and representative result. Secondly, *external validity* refers to the approximate truth of conclusions that involve generalizations. As for the exploratory interviews and validation interviews, we combined the gathered results of the scientific field with the gathered results of the business field. Also, for the survey we involved organizations that differ in several areas (by using the situational factors). Finally, we also tried to verify the results of the validation process with references from the literature before we applied them. The last threat, *reliability* concerns the demonstration that the results of the study can be replicated. For the business case, we covered this by creating an approach for the implementation of the productization process. Additionally, all other relevant information and conclusions are documented so that the results can be replicated.

5 Discussion

Based on the evaluation described in section 4, we feel that more validation is required to prove the added value of the productization process and to determine the validity and applicability of the identified stages. For example, there still exists some discussion on the identified stages of the productization process. Some members of the expert panel proposed to merge two stages and others commented on the fact that the end stages of the productization process are limited to customizable software products and standard software products (stage 6a and 6b). Furthermore, some experts mentioned that it could well be possible to use one of the other stages as end stage. In addition, some suggested that there should be a possibility to move backwards in the productization process, since there is always a possibility that an organization decides to change back to a customer specific focus. Additional case studies in different types of software companies should be carried out to validate the productization process in an organization that is transforming.

Another issue that needs to be discussed is the parallels that can be drawn between the productization approach and the evolution of product lines as described in, for example, [22]. In this article, Bosch describes the evolution of a product line through different maturity levels, ranging from independent products, via a shared product platform, to a configurable product base. Similarities with the productization process, of which the stages range from independent projects, via a shared product platform, to a standard software product, are easy to recognize. However, there are some important differences. The aim of software product line evolution is to transform the way in which products are being built. By using a configurable product base to develop a variety of products, the cost of development and the time to market can be decreased [22], compared to developing products from scratch. In the productization process, the aim is not directly to decrease costs and time to market, but to change the whole business model. Instead of carrying out customer-specific software projects, a shift is made to developing a standard software product for a market, hence serving a much larger array of customers. Nevertheless, the analogy with product line evolution is present.

6 Conclusions and Further Research

The research question, as we stated in Section 1, is:

How can organizations transform from developing customer-specific software to product software?

To answer this question, we proposed the productization process that describes the transformation from developing customer-specific software to product software and that can assist organizations in becoming a product software business. The entire productization process is supported by the implementation of the product management functions of the reference framework for SPM. Additionally, per stage a brief description is presented which explains in more detail the characteristics of that particular stage. The stages represent the different phases that can occur during such a transformation. Eventually, when an organization has reached its desired end stage, all product management functions should be in place.

We believe the productization process is a valuable contribution in the research domain of software business. However, this research also identifies the need for more research on the transformation from customer-specific software development to product software development.

Future research and more expert interviews or case studies are required in order to be able to provide a valid answer. Finally, research carried out by [2] also identified the overlap of Open Source software. Further research should look deeper into the integration of Open Source software in the productization process.

References

1. Sawyer, S.: Packaged software: implications of the differences from custom approaches to software development. European Journal of Information Systems 9, 47–58 (2000)
2. Xu, L., Brinkkemper, S.: Concepts of product software: Paving the road for urgently needed research. In: Castro, J., Teniente, E. (eds.) PHISE 2005, pp. 523–528. FEUP Press (2005)
3. Regnell, B., Höst, M., Natt och Dag, J.: An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software. Requirements Engineering 6, 51–62 (2001)
4. Weerd, I.v., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. In: International Workshop on Software Product Management (IWSPM 2006 - RE 2006), pp. 312–315. Minneapolis/St. Paul, Minnesota (2006)
5. Artz, P., Weerd, I.v., Brinkkemper, S.: Productization: An exploratory research project Technical report UU-CS-2010-003. Institute of Computing and Information Sciences, Utrecht University, http://www.cs.uu.nl/research/techreps/UU-CS-2010-003.html
6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly: Management Information Systems 28, 75–105 (2004)
7. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems (2007), AISWorld Net http://home.aisnet.org/displaycommon.cfm?an=1&subarticlenbr=279
8. Hietala, J., Kontio, J., Jokinen, J., Pyysiainen, J.: Challenges of Software Product Companies: Results of a National Survey in Finland. In: 10th International Software Metrics Symposium, pp. 232–243 (2004)
9. Hoch, D., Roeding, C.R., Purkert, G., Lindner, S., Müller, M.: Secrets of Software Success: Management Insights from 100 Software Firms around the World. Harvard Business School Press, Boston (1999)
10. Cusumano, M.A.: The business of software: What every manager, programmer and entrepreneur must know to thrive and survive in good times and bad. Free Press, New York (2004)
11. Keil, M., Carmel, E.: Customer-developer links in software development. Communications of the ACM 38, 33–44 (1995)
12. Carlshamre, P.: Release Planning in Market-Driven Software Product Development Provoking and Understanding. Requirements Engineering 7, 139–151 (2002)
13. Meyer, M.H., Seliger, R.: Product platforms in software development. Sloan Management Review 40, 61–74 (1998)
14. Codenie, W., De Hondt, K., Steyaert, P., Vercammen, A.: From Custom Applications to Domain-Specific Frameworks. Communications of the ACM 40, 71–77 (1997)
15. Bohl, O., Frankfurth, A., Schellhase, J., Winand, U.: Guidelines - A Critical Success Factor in the Development of Web-Based Trainings. In: International Conference on Computers in Education (ICCE 2002), Auckland, New Zealand, pp. 545–546 (2002)
16. Sandelowski, M.: The call to experts in qualitative research. Research in Nursing & Health 21, 467–471 (1999)
17. Zelkowitz, M., Wallace, D.: Experimental Models for Validating Technology. Computer 31, 23–31 (1998)

18. Bekkers, W., Weerd, I.v., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: Proceedings of the 2008 Second International Workshop on Software Product Management, pp. 41–48 (2008)
19. Weerd, I.: v., Bekkers, W., Brinkkemper, S.: Developing a Maturity Matrix for Software Product Management. UU-CS (Int. rep. 2009-15). UU WINFI Informatica en Informatiekunde
20. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Syed, M.R., Syed, S.N. (eds.) Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 28–38. Idea Group Publishing, Hershey (2007)
21. Yin, R.: Case study research: Design and methods, vol. 3. Sage Publishing, Beverly Hills (2003)
22. Bosch, J.: Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization. In: Chastek, G.J. (ed.) SPLC 2002. LNCS, vol. 2379, pp. 247–262. Springer, Heidelberg (2002)