

Developing a Maturity Matrix for Software Product Management

Inge van de Weerd, Willem Bekkers, and Sjaak Brinkkemper

Department of Information and Computing Sciences, Utrecht University,
Padualaan 14, 3584CH Utrecht The Netherlands
{i.vandeweerd,bekkers,s.brinkkemper}@cs.uu.nl

Abstract. The quality of processes in Software Product Management (SPM) has a high impact on the success of a software product, as it improves product quality and prevents release delays. To improve the SPM practice, we propose the maturity matrix for SPM, a focus area oriented maturity model concentrating on the SPM functions Requirements Management, Release Planning, Product Roadmapping, and Portfolio Management. In this paper, we describe the development of the SPM maturity matrix, consisting of (a) identification and description of capabilities, (b) positioning the capabilities at the right levels in the maturity matrix and (c) validating the maturity matrix with expert validation and a survey among 45 product managers and product management experts. The result is a validated maturity matrix that will guide further development of methodical support in SPM.

Keywords: software product management, maturity, requirements management.

1 Maturity in Software Product Management

Software product management (SPM) is a crucial area within many software companies. Good product management has a high impact on the success of a software product [1]. This requires a combination of technological, managerial and business skills, such as calculating optimal releases, setting out roadmaps, managing risks, and interacting with many internal and external stakeholders. If these activities do not get enough attention, the quality of a product decreases, release dates are not met, and managing customers' expectations become a large problem.

Although the product manager's function is highly important in the product software industry, little education exists in this area [2]. Almost no education on SPM is being offered, except in the area of marketing and sales. Based on our experience in the market, especially in The Netherlands, Germany and Switzerland, we observe that most software product managers were earlier employed in functions such as development manager, project manager or sales manager. This causes a gap of knowledge that the product manager has to solve by getting experienced in the area. Hence, lifting the quality of the product by improving the SPM processes is often difficult. Most existing software process improvement (SPI) models aim at a broad spectrum of SPI and the area of SPM is usually not the main area of attention.

In this research, we propose a maturity matrix for SPM that can be used to assess an organization’s current SPM capabilities and offer local, incremental improvements to the product manager.

In earlier research, we developed the Reference Framework for Software Product Management [2]. Since its publication, various studies have been done to test the reference framework in product software companies (cf. [3] and [4]). In this research, we use the reference framework as a foundation for our maturity matrix. Therefore, we will provide a brief explanation of the framework.

In Figure 1, the reference framework for software product management is depicted. The framework consists of *internal stakeholders* (product management, company board, sales & marketing, services, support, development and research & innovation) and *external stakeholders* (the market, partners and customers).

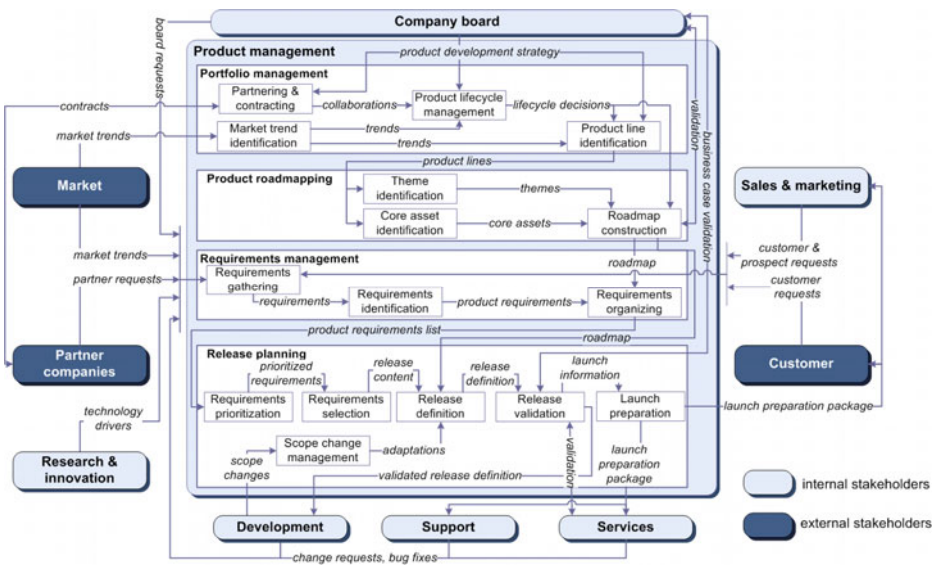


Fig. 1. Reference framework for Software Product Management

The most important internal stakeholder, Product management, consists of four business functions: (1) *Portfolio management* concerns managing the different products that a company owns. Partnering, product lifecycle management and product line identification are part of this function. (2) *Product roadmapping* handles with the development of the product roadmap, in which future releases are planned based on themes and core assets. (3) *Requirements management* contains the activities of requirements gathering, identification and organizing; all ongoing activities within the product management domain. (4) *Release planning* deals with prioritizing and selecting requirements in order to define the new release. Also the activities release validation, launch preparation, and scope change management are part of this function.

2 Research Design

This study follows the design science methodology, in which research is done through the processes of building and evaluating artifacts [5]. The artifact in this research is the maturity matrix for SPM. During our research, we follow the 5 process steps of the design cycle [6]. This design cycle consists of several steps that follow an iterative process; knowledge produced in the process by constructing and evaluating the artifact is used as an input for a better awareness of the problem. The 5 process steps are: (1) *Awareness of the problem*. In Section 1, we described the problem and its context. (2) *Suggestion*. The suggestion for a solution to the problem identified in step 1 is developed in this step. In Section 2, we describe our approach in tackling the problem and the research methods that we use. (3) *Development*. The development of the artifact, in this case the maturity matrix is described in Section 3. (4) *Evaluation*. This step comprises the evaluation of the method. We used a survey to validate the method, as is described in Section 4. The results of this survey lead to a higher level of problem awareness and suggestions for solutions. (5) *Conclusion*. Finally, in Section 5, conclusions and areas for further research are covered.

In Fig. 2, we depict the structures of the two artifacts in this research. The REFERENCE FRAMEWORK consists of KEY PROCESSES that are grouped into BUSINESS FUNCTIONS. Secondly, the MATURITY MATRIX consists of KEY PROCESSES and SPM CAPABILITIES. Each SPM CAPABILITY contributes to a KEY PROCESSES and it indicates which MATURITY LEVEL this process has.

In addition to the artifact structure, the research methods used during the development of both artifacts are provided. At the left, the research methods that were used for developing the Reference framework are listed and at the right the method for developing the Maturity matrix are listed.

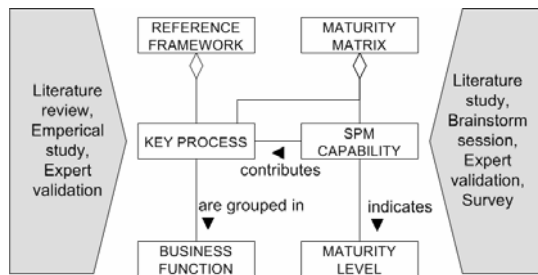


Fig. 2. Research methods and artifacts

As shown in Fig. 2, this research was conducted with the following data collection methods:

Literature study. One of the sources for the capabilities, which are defined for each of the processes in the reference framework for SPM, is a literature study.

Brainstorm session. A brainstorm session was conducted with experts from the scientific community to create the model. The session consists of two parts: 1) the capabilities themselves were determined; 2) the positions of the capabilities in the SPM

maturity matrix were determined. The literature study was used as a basis for the brainstorm session.

Expert validation. An expert from practice validated the results of the brainstorm session: the capabilities themselves and their position within the SPM Maturity Matrix.

Survey. A final validation was conducted based on a survey with SPM experts from practice from all over the world. The goal of this survey was to validate the order of the capabilities relative to each other in the SPM Maturity Matrix.

3 Developing a Maturity Matrix

In this section, we first describe our choice for the type of maturity matrix we use, and then we describe its structure and the development process.

3.1 Variance of Maturity Models

Van Steenberg et al. [10] recognize three variants of maturity models: (1) *staged 5-level models*, which distinguish five levels of maturity, which in turn have a number of focus areas that are defined specific to that level; (2) *continuous 5-level models*, which contain a number of focus areas, in each area the 5 levels are distinguished; and (3) *focus area oriented models*, in which each focus area has its own number of specific maturity level.

Most well-known maturity models are staged or continuous 5-level models, such as the Capability Maturity Model (CMM) [11], and its follow-up CMMI [12]. Earlier research into the improvement of SPM shows some shortcomings in these methods. CMMI for example, has been found too heavy to use by several organizations [13]. And there are others who say that extensive software process improvement (SPI) frameworks, such as CMMI and ISO/IEC 15504 (SPICE) [14] are too large to implement, or even comprehend [15] [16]. For example, a typical CMM SPI cycle can take between one and a half and two years to complete. It also requires large resources and long term commitment [17], which can be a problem for small and medium companies. Another problem is that small and medium software companies often not only lack the funds required to implement many of the practices from CMM but also have to base their SPI initiatives on practices that do not apply to them [18].

For the reasons above, we choose to develop a focus area oriented model, in order to make local analysis and incremental improvement possible. Similar model have been used for the testing domain [19] and the architecture domain [10].

3.2 Structure of the Maturity Matrix

In Table 1, the SPM maturity matrix is depicted. The matrix consists of columns and rows, which represent the two dimensions of the model. The columns 0 to 12 represent the different maturity levels for the model, where 0 is the lowest level of maturity, and 12 the highest. The SPM processes, or focus areas as they are called in the matrix, are represented by the rows and are divided into four groups (the business functions; ‘Requirements management’, ‘Release planning’, ‘Product roadmapping’,

‘Portfolio management’). When a focus area is carried out at a certain maturity level it is called a capability. In Table 1, we can for example identify four capabilities: A, B, C and D. Capability A, located on level 1, is coded as RM1A and described as: “Ad hoc requirements gathering. Requirements are being gathered and registered.”

Two more concepts need to be introduced:

Intra-process capability dependency. This is the dependency of one capability within a certain key process to another capability in the same key process. In Table 1 this type of dependency is depicted with an arrow between RM1B and RM1C.

Inter-process capability dependency. Intra-process refers to the dependency of a capability in a certain key process to a capability in another key process. In Table 1 this is depicted with an arrow between RM1D and RM3C.

A more detailed explanation on the structure of the matrix can be found in [7].

Table 1. Maturity Matrix Structure

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering		A		B	←	→	C			D			
RM2. Requirements identification			A			B			C				
RM3. Requirements organizing			A				B				C		
RP1. Requirements prioritization				A		B	C			D			
RP2. Requirements selection				A		B	C				D		
RP3. Release definition			A			B		C					
RP4. Release validation				A			B			C		D	
RP5. Launch preparation		A				B					C		
RP6. Scope change management				A		B				C			
PR1. Theme identification			A		B		C						
PR2. Core asset identification						A			B				C
PR3. Roadmap construction			A			B				C			
PM1. Market trend identification			A		B				C				
PM2. Partnering & contracting			A				B		C				
PM3. Product lifecycle management			A			B			C				
PM4. Product line identification				A			B			C			

3.3 Developing a Maturity Matrix for Software Product Management

We followed three main steps during the development of the maturity matrix:

1. Identification and description of capabilities

First, a literature study was carried out. This literature study was based on a multitude of papers describing specific processes within the field of SPM, e.g. [8] and [9]. Then, a brainstorm session with four SPM experts was held to identify the capabilities. Two of them had extensive professional experience in SPM and the other two were researchers to SPM.

2. Positioning the capabilities in the maturity matrix

The next step concerned positioning the capabilities in the matrix. By analyzing the inter- and intra-process capability dependencies, we decided the order of the

capabilities in the matrix. For example, for the first capability in the process in Requirements identification (RM2A), it makes sense to have gathered and registered requirements, which is capability RM1A. Therefore, RM2A must be placed at least 1 level after RM1A. This activity resulted in a matrix of 13 levels (0-12). During the validation, we will find out whether this is the right size.

3. *Validating the maturity matrix*

The third and last step is the empirical validation of the matrix. In Section 4, we describe how we use a survey to validate the positions of the capabilities.

4 Survey

Our survey consists of three parts: Introduction questions, general questions and capability questions. It starts with two introduction questions: “Which SPM areas are you familiar with?” and “How are you related to SPM?” The answers to these two questions determine which questions will be presented in the remainder of the survey. First, the respondent can choose which of the four SPM areas will be included in the survey. Only SPM-areas of which the checkboxes are ticked will be included in the survey. The second question is used to find out whether the respondent is a software product manager or another SPM professional. After the introduction question, some general questions are posed concerning company size, experience, etc.

The main structure of our survey is based on the four business functions that are defined in the reference framework for SPM: requirements management, release planning, product roadmapping, and portfolio management. For each function, we asked how our identified capabilities should be implemented in an organization. If we would ask to fill in a whole matrix per area, we would get very large matrices. For example, the release planning area has 21 capabilities and 12 rows. During the first pilots it appeared this would cause a cognitive workload that was too high. Therefore, we decided to use another approach and divided the matrix in three sub matrices. In the first matrix, ranging from 1 to 6, capabilities A and B are covered. In the second matrix, ranging from 4 to 9, capabilities B and C are covered. Finally, if necessary, a third matrix is used to cover capabilities C and D. In Table 2, part of the matrix is depicted, showing the division in three separate matrices.

Table 2. Matrix Division

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering													
RM2. Requirements identification													
RM3. Requirements organizing													

In Figure 3, we depict part of the survey in which all capabilities A and B of the Requirements Management process area are listed. The capabilities are listed on the left and the levels (1-6) in the middle. The last columns can be used to indicate whether the respondent has implemented the capability. The N/A option is for non-product managers that execute the survey.

	Maturity levels						Implemented?		
	1	2	3	4	5	6	Yes	No	N/A
Requirements are being gathered and registered.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For incoming requirements, immediately a consideration is made whether they can be implemented in the product.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For each release, requirements are classified by, for example, theme, function or core asset.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
All incoming requirements are stored in a central place (for example in a spreadsheet or tool).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Incoming requirements are being identified as market or product requirement. Product requirements are written in a pre-defined template.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The product roadmap is used as a guide in organizing and classifying requirements. Dependencies between different requirements are also identified.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Fig. 3. Survey Questions Requirements Management

4.1 Data Collection

In order to find respondents to the survey, we posted a message to several mailing lists. Below, you can find an overview of the members of these mailing lists:

- Netherlands product software mailing list: 175
- Netherlands SPM community: 68
- International SPM community: 176

In addition, during a national meeting among 20 Dutch product management professionals, the survey was promoted. Finally, the URL of the survey was posted on the SPM community website [20]. Please note that the members of the different mailing lists, visitors to the meeting, and visitors of the website overlap.

The number of respondent to the survey is 84. Of these 84 only 45 were useful, i.e. they did not quit the survey after the first two pages. Not all questions have the same amount of responses. The respondents can make a choice in the beginning of the survey, for which they can indicate on which SPM area they can answer questions. Per SPM area, we have the following amounts of respondents:

- Requirements management: 42
- Release planning: 27
- Product roadmapping: 25
- Portfolio management: 19

The respondents originate mostly from the Netherlands (36). Other individuals originate from Germany, Canada, USA, India, South Africa, Sweden, Switzerland and Spain. 13 of the respondents followed a course or study in SPM and three are certified in SPM. In Table 3, some characteristics concerning the respondents' function, company size (only product managers included) and experience are provided.

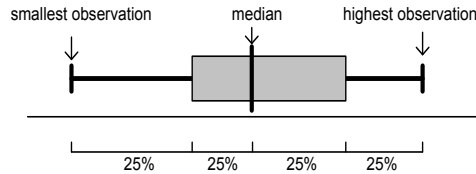
Table 3. Respondents' Characteristics

Function		Company size		Experience	
Product manager	30	Less than 50	12	0-2 years	13
Researcher	5	50 - 250	7	3-5 years	12
Consultant	3	More than 250	11	6-10 years	13
Other	7			More than 10 years	7

4.2 Data Analysis

In this section, we will analyze the results from the survey. For each business function, we describe the results of the survey and compare this with our initial maturity matrix. In principle, we use the survey results to update our maturity matrix. However, we make one exception. Dependencies between capabilities in the different business functions were not part of the survey. Therefore, in case a capability is placed on a certain maturity level because of a dependency to a capability in another business function, we keep to the original sequence.

We use box plots to give a graphical overview of the distribution of each capability, as is illustrated in Figure 4.

**Fig. 4.** Box plot

The box plots show the median, smallest and highest observations and the distribution over the quartiles. In addition, the outliers (if present) are shown. We use the medians as an indication of the maturity level of a capability, because we want to know which level was chosen most for a certain capability.

We list all capabilities within one business function in a box plot. At the Y-axis, the different capabilities are listed and at the X-axis, the maturity levels are indicated. Please note that some medians are not integers, but rational numbers. E.g. RM1C has a median of 5.5. The reason for this is that the respondents had to enter these maturity levels twice (cf. Section 4.1). Sometimes this resulted in two different answers, of which we calculated the mean. In such cases, we look at the distribution of the box plot to identify at which level the capability should be placed (in RM1C that means that the capability is placed on maturity level 6).

4.2.1 Requirements Management

In Fig. 5, the results of the Requirements management survey part are depicted in a box plot. In Table 4, the matrix is illustrated with the original results, and the deviations that were found when comparing it with the box plot. We also indicate the status

of a capability. “A” means that the result of the survey was the same as the original position, “A” indicates the result of the survey that was different from the original capability, and “A” shows the old position of the capability.

Several issues are noteworthy. Firstly, the intra-process capability dependencies are the same; all capabilities are positioned in the sequence A, B, C, (D). For the inter-process capability dependencies we see some differences that we will implement in the final matrix.

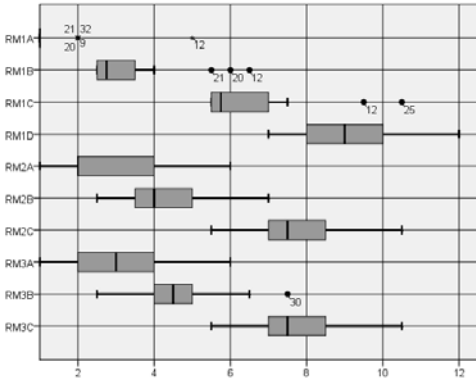


Fig. 5. Box plot Requirements Management

Table 4. Requirements management matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1		A		B			C			D			
RM2			A		<u>B</u>	<u>B</u>			C				
RM3			<u>A</u>	<u>A</u>		<u>B</u>	<u>B</u>		<u>C</u>		C		

4.2.2 Release Planning

In Fig. 6 and Table 5, the results of the Release planning part of the survey are illustrated and compared with the original matrix. Again, the intra-process capability dependencies are the same. However, in the inter-process capability dependencies we see many small deviations. All of which have been incorporated into the matrix. One issue stands out: The prioritizing and selection capabilities (RP1A and RP2A) are

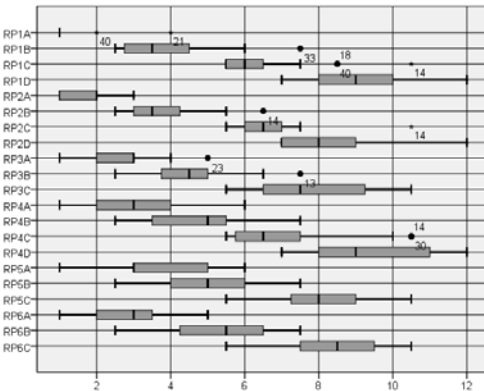


Fig. 6. Box plot release planning

Table 5. Release planning matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
RP1		<u>A</u>		<u>A</u>	<u>B</u>	<u>B</u>	C			D			
RP2			<u>A</u>	<u>A</u>	<u>B</u>	<u>B</u>		C	<u>D</u>		D		
RP3			A	<u>A</u>	<u>B</u>	<u>B</u>		C	<u>C</u>				
RP4				A		<u>B</u>	<u>B</u>	<u>C</u>		<u>D</u>		D	
RP5		A		<u>A</u>		B			C				
RP6				A		B				C			

implemented earlier than all other capabilities. Apparently, the respondents consider these activities as the minimum that a product manager should implement.

4.2.3 Product Roadmapping

The results of the Product roadmapping part of the survey are illustrated in Fig. 7 and compared with the original matrix in Table 6. Also for Product roadmapping, the intra-process capability dependencies are analogous to the original matrix. In the inter-process capability dependencies we see some deviations that will be included in the matrix.

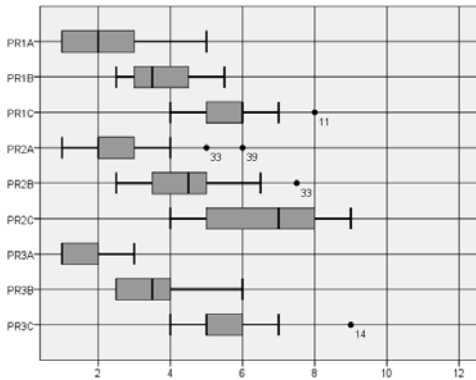


Fig. 7. Box plot product roadmapping

Table 6. Product roadmapping matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
PR1			A	B	C								
PR2			A	B	B	C							C
PR3		A	A	B	C					C			

4.2.4 Portfolio Management

Finally, in Fig. 8 and Table 7, the results of the Portfolio management part of the survey are illustrated and compared with the original matrix. Again, the intra-process capability dependencies are the same as in the original matrix. In the inter-process capability dependencies we see a few small deviations, which we will include in the matrix.

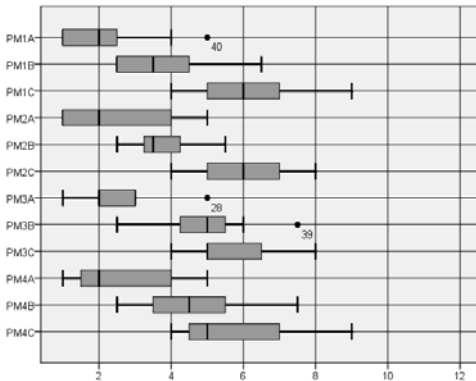


Fig. 8. Box plot portfolio management

Table 7. Portfolio management matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12
PM1			A	B	C				C				
PM2			A	B	C				C				
PM3			A		B	C			C				
PM4			A	A	B	C				C			

4.3 Results

In Table 8, the final maturity matrix for SPM is presented. We made some adaptations compared to the survey results. First, all capabilities of the Product roadmapping business function have all been shifted one level to the right. The dependencies between business functions were subject to the survey and we believe that the product roadmapping capabilities should be on a higher level. Finally, several capabilities that are placed on the highest maturity level of a certain area have been shifted to the right, namely RP2D, RP4D, PR2C and PR3C and all the C-capabilities of the Portfolio Management function. Reason for this is that we do not only want to incorporate the fastest way to implement the capabilities in our matrix, but we also would like to provide a balanced way to improvement.

In Table 8, we also show the score for one of the respondents in the benchmark questions. This respondent works for a company with less than 50 employees from the Netherlands. He did not have any education or certification in SPM. The overall SPM maturity level is determined by checking for which maturity level all capabilities (and the preceding capabilities) are implemented. As can be seen in the table, the maturity level in this case is 3. To advance to a higher maturity level, say level 4, work needs to be done in RM2, RP3, PM1 and PM2.

Table 8. Maturity matrix for Software Product Management

	0	1	2	3	4	5	6	7	8	9	10	11	12
RM1. Requirements gathering		A		B			C			D			
RM2. Requirements identification			A		B				C				
RM3. Requirements organizing				A		B			C				
RP1. Requirements prioritization		A			B		C			D			
RP2. Requirements selection			A		B			C			D		
RP3. Release definition				A	B				C				
RP4. Release validation				A		B		C				D	
RP5. Launch preparation				A		B			C				
RP6. Scope change management				A		B				C			
PR1. Theme identification				A		B		C					
PR2. Core asset identification				A		B							C
PR3. Roadmap construction			A		B						C		
PM1. Market trend identification			A		B			C					
PM2. Partnering & contracting			A		B			C					
PM3. Product lifecycle management			A			B		C					
PM4. Product line identification			A			B		C					

4.4 Threats to Validity

Validity refers to the degree to which a response measures what we intend for it to measure [21]. We identified three types of validity threats that are important to discuss, namely threats to conclusion validity, construct validity, and external validity. Internal validity is less important since this study is not about establishing a causal relationship. It is a descriptive study, in which we describe the mostly used implementation sequence of SPM capabilities.

Firstly, conclusion validity is the degree to which conclusions we reach about relationships in the data are reasonable. Important is the composition of participants and the statistical analysis. At the beginning of the survey, respondents were asked to indicate with which of the SPM areas they were familiar. The respondents only got to answer questions about these areas. Therefore, we believe that the respondents were experienced enough to answer the questions they selected. Most of the respondents are not anonymously, since an email address is needed to send them the benchmark report. However, there is no reason to believe that they have not answered according to their best knowledge, since there is no social pressure to answer in a particular way. Therefore, we consider the conclusion validity not to be critical.

Construct validity concerns whether we measure the construct that we believe we measure. On the introduction page of the survey, we have provided all definitions on all important concepts used in the questionnaire. Hence, we minimized the threat that participants interpret concepts differently. However, it is not possible to completely eliminate this threat.

External validity concerns generalization of the results to other groups and contexts than the one studied. The survey was sent to different groups of SPM professionals, as is described in Section 4.2. The respondents are mainly product managers with varying experience, working for companies of varying sizes. Most of the respondents are from the Netherlands. A drawback is that only 46 respondents filled out the survey. However, given the diverse background of the respondents concerning company size and experience, we believe that the validity is high within the product manager population in the Netherlands. Since we believe SPM is not particularly influenced by culture, we believe that we can also generalize the results to other countries. To be sure, more research is needed.

5 Conclusions and Future Research

In this paper, we described the development of a maturity framework for SPM that can be used to assess an organization's current SPM capabilities and offer local, incremental improvements to the product manager. After developing the matrix, we used a survey in which 45 SPM professionals reported by indicating the right order in which SPM capabilities should be implemented in an organization. The results of this survey supported our initial positioning of the SPM capabilities, at least, when looking at the inter-process capability dependencies. Concerning the intra-process capability dependencies, several deviations were found, of which most of them have been used to improve the matrix. Currently, we are carrying out multiple case studies to refine the capabilities.

We believe that the maturity matrix for SPM is a valuable tool for process assessment in SPM. In addition, the matrix can also be used as a means to achieve process improvement, for example by using it in the Product Software Knowledge Infrastructure (PSKI) [22], which creates a process advice for product software companies by taking its situational factors into account [23]. In future research, we will carry out case studies at different product software companies to further validate and refine the matrix in SPM capability improvement.

References

1. Ebert, C.: The Impacts of Software Product Management. *Journal of Systems and Software* 80(6), 850–861 (2007)
2. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. In: *Proceedings of the 14th Int. Requirements Engineering Conference, Minneapolis/St. Paul, USA*, pp. 312–315 (2006)
3. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Relevance of Situational Factors in Software Product Management. Technical report, UU-CS-2008-016, Utrecht University (2008)
4. van de Weerd, I., Brinkkemper, S., Versendaal, J.: Concepts for Incremental Method Evolution: Empirical Exploration and Validation in Requirements Management. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) *CAiSE 2007 and WES 2007*. LNCS, vol. 4495, pp. 469–484. Springer, Heidelberg (2007)
5. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. *MIS Quarterly* 28, 75–105 (2004)
6. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems, AISWorld Net: <http://www.isworld.org/Researchdesign/drisISworld.htm> (retrieved December 7, 2007)
7. van Steenberg, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: The Design of Focus Area Maturity Models. In: *Proceedings of DESRIST 2010: Global Perspectives on Design Science Research, St.Gallen, Switzerland, June 4-5 (2010)*
8. Abramovici, M., Soeg, O.C.: Status and Development Trends of Product Lifecycle Management Systems. Ruhr-University Bochum, Chair of ITM, Germany (2002)
9. Clements, P., Northrop, L.: *Software Product Lines: Patterns and Practice*. Addison Wesley, Reading (2001)
10. Van Steenberg, M., Brinkkemper, S., van den Berg, M.: An Instrument for the Development of the Enterprise Architecture Practice. In: *Proceedings of the 9th Int. Conference on Enterprise Information Systems*, pp. 14–22 (2007)
11. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: *Capability Maturity Model, Version 1.1*. IEEE Software 10(4), 18–27 (1993)
12. CMMI Product Team: *Capability Maturity Model Integration (CMMI) Version 1.1 Staged Representation*, Tech. Report CMU/SEI-2002-TR-029, Carnegie Mellon University (2002)
13. Cusumano, M.A.: *The Business of Software*. Free Press, New York (2004)
14. ISO/IEC 15504, *Information Technology – Software Process Assessment ISO/IEC (1996)*
15. Kuilboer, J.P., Ashrafi, N.: Software Process and Product Improvement: An Empirical Assessment. *Information and Software Technology* 42(1), 27–34 (2000)
16. Reifer, D.J.: The CMMi: It's Formidable. *Journal of Systems and Software* 50(2), 97–98 (2000)
17. Zahran, S.: *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley, Reading (1997)
18. Brodman, J.G., Johnson, D.L.: What Small Businesses and Small Organization Say About the CMM. In: *Proceedings of the 16th International Conference on Software Engineering*, pp. 247–262. IEEE Computer Society Press, Los Alamitos (1994)
19. Koomen, T., Baarda, R.: TMap Test Topics, Tutein Nolthenius, Den Bosch, The Netherlands (2005)
20. van de Weerd, I.: *IT and Software Product Management*, <http://www.softwareproductmanagement.org/> (retrieved Februari 1, 2009)

21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell (1999)
22. van de Weerd, I., Versendaal, J., Brinkkemper, S.: *A Product Software Knowledge Infrastructure for Situational Capability Maturation: Vision and Case Studies in Product Management*. In: *Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2006)*, Luxembourg, pp. 97–112 (2006)
23. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: *The Influence of Situational Factors in Software Product Management: An Empirical Study*. In: *Proceedings of the 21st International Workshop on Product Management, Barcelona* (2008)