A SITUATIONAL ASSESSMENT METHOD FOR SOFTWARE PRODUCT MANAGEMENT

Bekkers, Willem, Utrecht University, Padualaan 14, Centrum Gebouw Noord, 3508 TB Utrecht, NL, bekkers@cs.uu.nl

Spruit, Marco, Utrecht University, Padualaan 14, Centrum Gebouw Noord, 3508 TB Utrecht, NL, m.r.spruit@cs.uu.nl

Weerd, van de, Inge, Utrecht University, Padualaan 14, Centrum Gebouw Noord, 3508 TB Utrecht, NL, i.vandeweerd@cs.uu.nl

Vliet, van, Rob, Centric, Antwerpseweg 8, 2803 PB Gouda, NL, rob.van.vliet@centric.nl

Mahieu, Alain, Centric, Antwerpseweg 8, 2803 PB Gouda, NL, alain.mahieu@centric.nl

Abstract

There is very little literature supporting software product managers in their work, even though they play a crucial role within software product management organizations. There are large solutions for the improvement of software product management practices, but these are not applicable to most small and medium sized organizations. This research-in-progress paper presents a general incremental assessment method that takes the organization's situational context into account: the Situational Assessment Method (SAM). We applied this scientific method to the field of software product management to solve the aforementioned problem. Our method presents organizations with an assessment of their current maturity level, and suggests steps to incrementally improve their processes. SAM is a focus area oriented instrument with a different set of capabilities for each area. The context of an organization is taken into account by examining various situational factors that describe the context of the organization, and the organization itself. This context is then used to determine which capabilities apply to the organization being assessed. A situation specific advice indicating how software product management practices can be improved upon is then created based on a gap analysis of the currently implemented capabilities, and the capabilities that should be implemented.

Keywords: Software product management, Assessment, Situational factor, Maturity model, Requirements management.

1. INTRODUCTION

1.1. Motivation and research goal

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 (Ebert, 2007). 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 becomes a large problem.

Earlier research showed that even though the product manager's function is highly important in the product software industry, little education exists in this area (Weerd, Brinkkemper, Nieuwenhuis, Versendaal, & Bijlsma, 2006). 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.

To aid product managers in improving their SPM practices, we propose the Situational Assessment Method (SAM). The SAM can be used to measure an organization's SPM maturity level and determine the areas that need improvement to reach a higher maturity level. This assessment method uses a SPM maturity model to determine which SPM capabilities are implemented in the organization and which SPM capabilities should be implemented in the organization. By analyzing the gaps, local, incremental improvements can be suggested to the product manager. The assessment method should require relatively little time and expenses to execute. Furthermore, all kinds of organizations, including small and medium sized organizations, should be able to use the assessment method. Finally, the model we create should be applicable in different areas of attention. The version described in this paper is aimed at SPM.

1.2. Research method

This study follows the design science methodology, in which research is done through the processes of building and evaluating artifacts (Hevner, March, Park, & Ram, 2004). The artifact in this research is the Situational Assessment Method (SAM) which we applied to the SPM domain. During our research we follow the five process steps of the design cycle (Vaishnavi & Kuechler, 2007). 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 five process steps are:

- **Awareness of the problem** – In section 1, we described the problem and its context.
- **Suggestion** – The suggestion for a solution to the problem identified in step 1 is developed in this step. In this section, we describe our approach in tackling the problem and the research methods that we use.
- **Development** – The development of the artifact, in this case the situational maturity assessment described in section 3.
- **Evaluation** – This step comprises the evaluation of the method. We used an expert validation, case studies, and 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. We show one of the case studies in section 4.
- **Conclusion** – Finally, in section 5, conclusions and areas for further research are covered.

During this research, we made use of several data collection sources. Firstly, we performed a *literature study*. This study was used as one of the sources for the capabilities, which are defined for each of the processes in the reference framework for SPM. The literature study was based on a multitude of papers describing specific processes within the field of SPM (e.g. Abramovici & Soeg (2002), Clements &

Northrop (2001)). Secondly, a *brainstorm session* was conducted with experts from the scientific community to create the model. The session consisted of two parts: 1) the capabilities themselves were determined; 2) the positions of the capabilities relative to each other were determined. The literature study was used as a basis for the brainstorm session. Furthermore, an *expert validation* was held where business professionals validated the results of the brainstorm session: the capabilities themselves, their order, and their weights. Finally, four *case studies* were performed at SPM companies from the Netherlands to test the applicability of the assessment method (see section 4) in day-to-day business environment.

2. RELATED WORK

The SAM is based on several key elements that we will describe in the following sub sections. First, we provide an overview of the different types of maturity models, and explain our choice of one of these maturity models. Then, we explain the reference framework for software product management that is used to structure the SAM. Finally, we explain how situational factors are used in the SAM.

2.1. Maturity models

Various types of maturity models have been described and used in science and business. Van Steenbergen et al. (2007) make the following categorization of maturity models:

- **Staged 5-level models** – Five levels of maturity are distinguished, which in turn have a number of focus areas that are defined specific to that level. The most well-known examples are the Capability Maturity Model (CMM) (Paulk, Curtis, Chrissis, & Weber, 1993), and its follow-up CMMI (CMMI Product Team, 2002)
- **Continuous 5-level models** – These models consist of a number of focus areas for which five levels are distinguished. Examples of these models are Appel (2000) and Westbrock (2004).
- **Focus area oriented models** – Each focus area has its own number of specific maturity level. These models are used in the testing domain Koomen & Baarda (2005) and the architecture domain Steenbergen & Brinkkemper (2007).

Earlier research into the improvement of SPM shows some shortcomings in the staged and continuous 5-level models. CMMI for example, has been found too heavy to use by several organizations (Cusamo, 2004). Furthermore, extensive software process improvement (SPI) frameworks, such as CMMI and ISO/IEC 15504 (SPICE) (ISO/IEC-15504, 1998) are too large to implement, or even comprehend (Kuilboer & Ashrafi, (2000), Reifer (2000)). 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, which can be a problem for small and medium companies (Zahran, 1997). 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 (Brodman & Johnson, 1994).

For the reasons above, we choose to develop a focus area oriented model, in order to enable local analysis and incremental improvement.

2.2. Software Product Management

In earlier research we developed the Reference Framework for Software Product Management (Weerd, et al., 2006). Various studies have been performed to test the reference framework in product software companies since its publication (cf. Bekkers & Weerd & Brinkkemper & Mahieu (2008), Weerd & Brinkkemper & Versendaal (2007)). In this research, we use the reference framework as a source for the focus areas in our maturity matrix. Therefore we will provide a brief explanation of the framework below.

Figure 1 depicts the reference framework for software product management. 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). The most important internal stakeholder, product management, consists of four business functions: portfolio management, product roadmapping, requirements management and release planning. Each of these business functions consists of a number of focus areas, as shown in Figure 1.

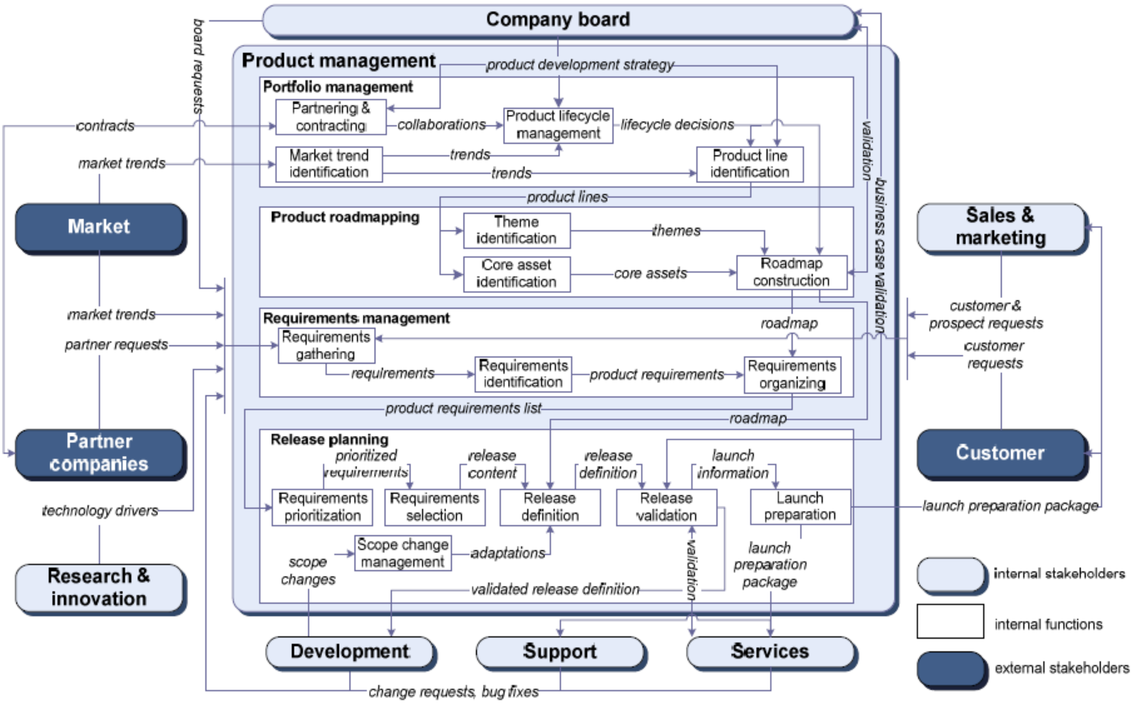

Figure 1. Reference framework for Software Product Management (Weerd, et al. 2006)

2.3. Situational Factors

A Situational Factor (SF) contains information about the process, the context of the organization, and the organization itself (Bekkers, 2008). SFs describe the situational context in which, in this case, the product manager has to operate and to which the SPM processes thus have to fine-tuned. An example of an SF in SPM is ‘Customer involvement’, which indicates to what extent a customer wishes to be involved in the SPM processes. When the value of this SF changes, some SPM processes may need to be changed to respond to the new environment. In previous research we presented a list of 27 SFs in five categories, relevant to SPM, with the level of influence they have on the selection of (parts of) methods (Bekkers, Weerd, Brinkkemper, Mahieu, 2008). These SFs are divided into five categories: organizational characteristics, customer characteristics, market characteristics, product characteristics, and stakeholder involvement. In the SAM, SF values will be used to determine situational context based maturity goals specifically for the organization being assessed.

3. SITUATIONAL ASSESSMENT METHOD

The SAM is an assessment method that can be applied as a general assessment method. We developed a variant for software product management: the Situational Assessment Method for Software Product Management (SAM-SPM).

The SAM consists of four components: knowledge base, questionnaire, calculation, and feedback (see Figure 2).

- **Knowledge base** – The knowledge base contains the knowledge on which the assessment advice is based: the *Capability Matrix* (CM), a matrix containing all capabilities within the field of SPM in an ordered manner, the *Situational Factors* (SFs) for SPM, and the *Situational Factor Effects* (SFEs), the effects certain specific SF values have on the capabilities.

- **Questionnaire** – The questionnaire consists of two separate questionnaires. The *Implemented Capabilities* questionnaire determines which capabilities are implemented within the organization. The *Situational context* questionnaire gathers the SF values for the organization.
- **Calculation** – The calculation determines, based on the input from the questionnaires, what the current maturity is, and what the optimal capabilities are for the organization being assessed. The current maturity is modeled in the *Current Capability Profile* (CCP), the optimal maturity is modeled in the *Optimal Capability Profile* (OCP). A comparison between the current and the optimal situation results in an overview of the problem areas that need improving, this is modeled in *Areas of Improvement Matrix* (AIM). The AIM is thus a custom-made advice for an organization.
- **Feedback** – The feedback consists of an evaluation that is performed to update the knowledge base. It can result in the addition, adjustment, or removal of knowledge components. The evaluation is performed after each assessment, but it can also be performed based on new scientific literature, case studies, or expert interviews which have not yet been incorporated in the knowledge base.

The four components above are explained further in the following four sections (section 4.1 to 4.4).

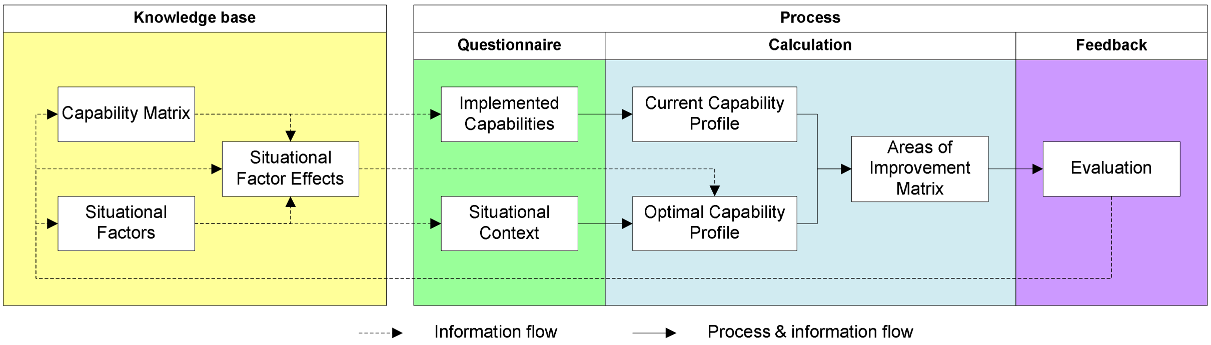

Figure 1. The four components of the Situational Assessment Method (SAM)

3.1. Knowledge base

A capability is an element contributing to the maturity within a specific focus area. It can be a process, the use of standards, or a technical instrument that needs to be implemented by an organization to reach a higher maturity level. An example of a process is 'Requirements prioritization', a standard can be a standard format in which requirements are recorded, and a technical instrument can be a central database for requirements. Table 1 shows a capability from the SAM-SPM.

We describe the following six attributes for the capabilities in the CM:

- **Name** – A name describing the capability in a few words.
- **Weight** – The weight indicates the amount of effort the capability costs to perform after it has been implemented. A scale ranging from low to high indicates the effort.
- **Goal** – The goal describes what purpose the capability serves, it indicates the advantage of executing the capability.
- **Action** – The action describes what must be done in order to meet the capability.
- **Prerequisite(s)** – Some capabilities require that one or more other capabilities be implemented first. This optional relation is described here by listing all the capabilities that have to be implemented first. There are two types of dependencies, both of which are indicated at the prerequisites. Firstly, we define intra-process capability dependency: this is the dependency of one capability within a focus area to another capability in the same focus area. And Secondly, we distinguish inter-process capability dependency: this type of dependency refers to a dependency of a capability in a focus area to a capability in another focus area.
- **Reference(s)** – The optional reference attribute describes related literature which can aid in understanding and implementing the capability, this literature thus has a supporting role.

Name	Requirement dependency linking
Weight	High
Goal	The existence of requirement interdependencies means that requirements interact with and affect each other. Requirement dependency linking prevents problems that result from these interdependencies, and therewith enables better planning of the development process.
Action	Dependencies between market and product requirements are determined and registered. A dependency exists when a requirement demands a specific action of another requirement. E.g. a requirement demands that another requirement be implemented too, or that another requirement is not implemented in case of conflicting requirements. The linkage can be supported by using advanced techniques, such as linguistic engineering.
Prerequisite(s)	RG:A
Reference(s)	Dahlstedt & Persson (2003)

Table 1: Capability C of the focus area Requirements organizing (RO:C) within the SAM-SPM

There are two generally applicable criteria that apply to all capabilities in order for them to reach the implemented status within the model. Firstly all capabilities must be reoccurring. This means that the process must be executed on a reoccurring and planned basis, and not ad hoc. The SAM-SPM model is intended to improve continuous processes, its capabilities therefore also describe reoccurring (or continuous) actions. If a capability is not executed on a regular, predetermined basis, then the capability is not satisfied within this model. Secondly, all capabilities must be documented. A detailed description of the processes must be described in a document for all parties involved in the capability. All parties involved in the capability must at least have access to the (part of) the process describing the actions that are required of them. These actions may be described in a less formal way so that the document is also understandable to external parties (e.g. customers).

Capabilities are labeled using the following guidelines: a combination is made of a two or three letter abbreviation indicating the process, a colon, and a letter corresponding to the capability within the process. The identifier for capability A of the 'Requirements gathering' process will thus be RG:A.

Focus area		**Maturity levels**												
Title	**Code**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Requirements management														
Requirements gathering	RG		A		B	C		D	E	F				
Requirements identification	RI			A			B		C		D			E
Requirements organizing	RO			A		B		C						
Release planning														
Requirements prioritization	RP				A		B	C	D		E			
Requirements selection	RS			A						B	C	D		
Release definition	RD				A	B	C		D		E			
Release validation	RV				A			B			C		D	
Launch preparation	LP		A				B		C			D		
Scope change management	SCM				A		B				C		D	
Product roadmapping														
Theme identification	TI					A		B						
Core asset identification	CAI						A			B				C
Roadmap construction	RC			A			B	C		D	E	F		
Portfolio management														
Market trend identification	MTI			A		B				C				
Partnering & contracting	PC			A		B		C		D				
Product lifecycle management	PLM			A			B			C				
Product line identification	PLI							A			B			

Table 2: The Capability Matrix (CM) from the Knowledge Base component.

One of the components in the SAM knowledge base is the Capability Matrix (CM). The CM provides an overview of all the capabilities that need to be implemented to reach a full-grown maturity (see

Table 2). The matrix consists of columns and rows, which represent the two dimensions of the maturity model. The SPM key processes are represented by the rows in the focus area column and are divided into four groups (the business functions: Requirements management, Release planning, Product roadmapping, Portfolio management). The columns 0 to 12 represent the maturity levels (where zero is low and twelve is high). The The letters A to F represent the capabilities. Each focus area has its own unique capabilities, the amount of capabilities within a focus area varies from two (A-B) to six (A-F). The CM suggests the best implementation order for the capabilities (from left to right). The placement of the capabilities is based on a series of interviews with experts from both the scientific world and the field of practice, and questionnaires among product managers.

The Situational Factor Effects (SFEs) provide a method to model product managers' knowledge. It reflects what should be done under certain circumstances (a specific SF value, or range of values). It does this by modeling what effect specific values, or value ranges, of SFs have on one or more capabilities. In this manner both situations where a capability does not need to be implemented because of a situational context (the SF values), as well as situations where a capability needs to be implemented because of a situational context, can be modeled. Conflicts are solved by making SFEs, which indicate that a capability does not need not be implemented, overrule SFEs that indicate that a capability needs to be implemented.

The current SFE knowledge base has populated with representations of the knowledge of experts, this knowledge was gathered during expert interviews. Four examples of these SFEs can be seen in Table 3. The first example is the SF 'Customer involvement', which represents the wish of the customer to be involved in the SPM processes. If this SF has the value 'Low', meaning that the customer does not want to be involved in the processes, then this SFE disables the capabilities that involve the customer in the SPM processes. The SFE for 'Customer variability' shows that a SFE can also enable a capability. Finally, the SFEs for 'Partner involvement' shows that a SF can have an effect on more than one capability.

Situational factor	**Operator**	**Value**	**Capability**	**Effect**
Customer involvement	=	Low	RG:F	Disable
Customer variability	>	10%	RI:C	Enable
Partner involvement	=	Low	RG:F	Disable
Partner involvement	=	Low	RP:C	Disable

Table 3: Four Situational Factor Effects (SFE) examples of the SAM-SPM

3.2. The questionnaires

There are two questionnaires in the SAM. The first one is the *situational context questionnaire.* This questionnaire determines the situational context of the organization. It consists of a series of SFs for which the organization has to indicate their own values. There are currently 26 questions in this questionnaire for the SAM-SPM (see Table 4 for two examples from the questionnaire).The second questionnaire is the *capability questionnaire*. This questionnaire determines which capabilities are implemented within the organization. It consists of one statement for each capabilities in the CM, the organization being assessed needs to answer the question 'Have you implemented this capability within your organization?' with either yes or no for each statement. There are currently 61 questions in this questionnaire for the SAM-SPM (see Table 5 for two examples from the questionnaire). Both questionnaires are close-ended, making them fast and easy to fill out for the organization being assessed.

The capability prerequisites (one of the attributes of the capabilities as defined in section 3.1) allow for the creation of an intelligent capability questionnaire. There is no need to ask the interviewee whether a capability has been implemented for which the mandatory prerequisites have not been met. Such a capability cannot be true. The number of questions that need to be asked can therefore be minimized by applying an intelligent ordering of the questions in the capability questionnaire.

Situational factor	Description	Unit	Answer
New requirements rate	The number of new feature requests per year from all sources (e.g. customers and sales).	Feature requests per year	60
Number of products	The number of other products in the product line for this product (this can thus be zero to many).	Number of products	1

Table 4: Two examples from the 'Situational Context' questionnaire of the SAM-SPM

Code	Statement	Answer
RI:B	The correctness ("Is the definition correct?") and completeness ("Does the requirement describe all relevant aspects?") of the requirement is validated.	No
SCM:C	An impact analysis is performed to determine the effects of the scope change.	Yes

Table 5: Two examples from the 'Implemented Capabilities' questionnaire of the SAM-SPM

3.3. The calculation

There are three steps in the maturity calculation. The first two steps can be executed in parallel, and serve as input for the third step.

First, the current capability profile (CCP) is determined based on the capabilities that are currently implemented within the organization. This profile can be deduced directly from the 'Implemented capabilities' questionnaire.

Secondly, the organizations SF values (indicated in the 'Situational context' questionnaire) are applied to the SFE rules (which determine which capabilities should be enabled in a situation). This results in the optimal capability profile (OCP), a custom CM tailored to the situational context of the organization. Note that some of the capabilities of the CM may be disabled in this tailored CM, because they are not relevant in situational context of the organization.

Finally, the improvement areas are determined. These are determined by comparing the CCP with the OCP. This results in a matrix detailing the differences between the currently implemented and the optimal set of capabilities, this matrix is called the Areas of Improvement Matrix (AIM). The AIM indicates the status of each capability, the different statuses are: 'implemented', for capabilities that need to be implemented, and are indeed implemented; 'missing', for capabilities that need to be implemented, but have not been implemented; N/A', for capabilities that need not be implemented based on the SFEs, and have not been implemented; and finally 'extra', for capabilities that need not be implemented but are implemented.

An organizations maturity level is determined in the same manner as Steenbergen, et al. (2007) present in their focus area oriented model, with the addition that we ignore capabilities that have been disabled by SFEs. The maturity level of the organization is the highest level for which all of its enabled capabilities, and the enabled capabilities of the previous levels have been satisfied by the organization. To determine the maturity level we can thus scan the CM left to right stopping the level before the level where a capability has not been satisfied. The level we stopped at will then be the current maturity level for the organization. This means that if the capabilities RG:A and LP:A are satisfied, and RG:B has not been satisfied, then the maturity level is 1.

3.4. Feedback

The field of SPM is constantly evolving. The content of the knowledge base will therefore need to evolve with the SPM field. It does this by performing an evaluation after each assessment. This evaluation is used as feedback to determine whether the capabilities, SFs, and SFEs are still correct and complete. This mechanism enables the knowledge base to evolve over time, becoming more complete and correct, and remain up-to-date with the field SPM. A first implementation of the knowledge base is currently in the early stages of development, based on expert interviews (both from the scientific and practical fields), case studies, and scientific literature. Expert interviews will be conducted to determine the SFEs.

4. SAM-SPM CASE STUDY

In this section we present results of a test case at a SPM organization where we tested the SAM-SPM. The case company is a small organization that only has a few customers. These customers are highly involved in the SPM processes. We only present the results of the 'Requirements management' and 'Release planning' focus areas due to the limited space available in this paper.

The results of the Implemented Capabilities questionnaire determine the CCP (see Table 6). The Situational Context questionnaire together with the SFEs determine the OCP (see Table 7). The CCP and OCP combined result in the AIM, this matrix shows where the problems, and therewith the opportunities for improvements, lie.

We will take capability D from 'Requirements identification' (RI:D) as an example. The statement for this capability in the situational context questionnaire is the following: 'Incoming requirements are identified as being either a market or product requirement. Market requirements are rewritten as product requirements, both requirement types have a pre-defined template.'. This statement is not true for the case company, thus the answer for RI:D in the CCP is 'No'. RI:D should be performed by default and is not disabled by any SFE based on the case company situational context, thus the answer for RI:D in the OCP is 'Yes'. Capability RI:D should be executed, but is not executed, it is therefore indicated as 'Missing' in the AIM.

Focus area		Capabilities					
Title	Code	A	B	C	D	E	F
Requirements gathering	RG	Yes	Yes	No	Yes	Yes	No
Requirements identification	RI	Yes	No	Yes	No	No	
Requirements organization	RO	No	Yes	No			
Requirements prioritization	RP	Yes	Yes	No	No	No	
Requirements selection	RS	Yes	Yes	No	No		
Release definition	RD	Yes	No	No	Yes	Yes	
Release validation	RV	Yes	No	No	No		
Launch preparation	LP	Yes	Yes	Yes	No		
Scope change management	SCM	Yes	No	Yes	Yes		

Table 6: Case company's Current Capability Profile (CCP) for the 'Requirements management' & 'Release validation'

Focus area		Capabilities					
Title	Code	A	B	C	D	E	F
Requirements gathering	RG	Yes	Yes	No	Yes	Yes	No
Requirements identification	RI	Yes	Yes	Yes	Yes	No	
Requirements organization	RO	Yes	Yes	Yes			
Requirements prioritization	RP	Yes	Yes	No	Yes	Yes	
Requirements selection	RS	Yes	Yes	No	Yes		
Release definition	RD	Yes	Yes	Yes	Yes	Yes	
Release validation	RV	Yes	Yes	Yes	Yes		
Launch preparation	LP	Yes	Yes	Yes	Yes		
Scope change management	SCM	Yes	Yes	Yes	Yes		

Table 7: Case company's Optimal Capability Profile (OCP) for the 'Requirements management' & 'Release validation'

The case company's maturity level (based on the focus areas presented in Table 6) is thus 1. Since they haven't implemented the mandatory capability A of the focus area 'Requirements organization' placed on maturity level 2.

If we look at capability RG:A (capability A within the focus area of ‘Requirements gathering’) in the CCP, than we can see that it has been implemented within the organization. The OCP tells us that the capability needs to be implemented to achieve optimal maturity. This capability will thus be indicated as ‘Implemented’ in the AIM. Capability RO:A is indicated as not being implemented in the CCP, while the OCP indicates that it should be implemented. This capability has thus been indicated as ‘Missing’ in the AIM. As a final example we look at capability RG:C which hasn’t been implemented and does not have to implemented, it is thus indicated as ‘N/A’ in the AIM.

Focus area	Capabilities					
	A	B	C	D	E	F
Requirements gathering	Implemented	Implemented	N/A	Implemented	Implemented	N/A
Requirements identification	Implemented	Missing[2]	Implemented	Missing[4]	N/A	
Requirements organizing	Missing[1]	Implemented	Missing[3]			
Requirements prioritization	Implemented	Implemented	Missing	Missing	Missing	
Requirements selection	Implemented	Implemented	Missing	Missing		
Release definition	Implemented	Missing	Missing	Implemented	Implemented	
Release validation	Implemented	Missing	Missing	Missing		
Launch preparation	Implemented	Implemented	Implemented	Missing		
Scope change management	Implemented	Missing	Implemented	Implemented		

Table 8: Case company’s Areas of Improvement Matrix (AIM) for the ‘Requirements management’ & ‘Release validation’

The resulting high-level advice for the focus area group of ‘Requirements management’ (codes: RG, RI, RO) for the case company is (1) to start organizing their requirements (capability RO:A), (2) validate requirements (capability RI:B), (3) map requirement dependencies (capability RO:C), and (4) model the requirements in a uniform manner (capability RI:D). The advice given is accompanied by a more detailed description of the capabilities, which also refers to relevant literature for the capability, and explains what the advantage is of implementing the capability (see Table 1 for an example of a complete capability description).

The case company is also advised to implement the capabilities in the following order: first RO:A, then RI:B, followed by RO:C, and finally RI:D. This order suggestion follows from the ordering within the CM (see Table 2) and the prerequisites that some capabilities have (indicating a mandatory order).

The case company recognized the advice as valid points for improvement.

5. CONCLUSIONS & FUTURE RESEARCH

5.1. Conclusions

We believe that the SAM is an effective method to aid organizations in the improvement of their processes (SPM processes in the case of the SAM-SPM). There are many advantages to using it. The SAM is *solution oriented.* It does not only indicate what the current maturity is, it also shows the organization where there is room for improvement by indicating the capabilities that need to be implemented. The SAM sets a *realistic goal* for organizations by setting goals based on their situational environment instead of treating all organizations the same, having them perform all the same capabilities.

The SAM allows for *incremental growth* as well as a big bang approach. An organization can choose which strategy to apply for its maturity improvement implementation. It can choose to implement new capabilities one at a time (incremental), or many in one step (big bang). This allows the organization to determine how much money and time it wants to invest in its maturity improvement.

The assessment requires relatively *little effort* of the organization being assessed. The organization only has to fill out two questionnaires: one questionnaire with a yes/no question for each capability, and one questionnaire with close-ended questions for each SF. This makes it quick and easy for the organization being assessed to provide the input needed in the assessment. The SAM can work fully *automatic* and can therefore present its results directly after the user has supplied the information. This results in fast, repeatable, and verifiable results.

It is possible to produce *results with partial input data*, though the results may provide a less customized advice. The method *can be applied partially* since it can produce results per focus area or per focus area group. These can thus be investigated without involving the other focus areas. This allows the organization to assess a specific aspect of its organization (e.g. Requirements gathering).

The SAM has been constructed to allow for future changes. It is possible to add, modify, or remove processes, capabilities, SFs, and SFEs, which makes it a *future proof* method.

5.2. Future research

The largest part of the future work consists of the gathering and validation of the various knowledge components in the SAM-SPM. The completeness and correctness of the gathered content in all knowledge base components also needs to be validated.

There are also opportunities for improvement of the SAM model by developing the intelligent questionnaires as discussed in section 3.2, and creating an advanced evaluation method to update the knowledge base.

It is not always possible to give a simple yes/no answer when determining whether the capabilities have been satisfied by the organization. E.g., an organization can put a certain capability into practice most of the time, but not always. Also, capabilities can be applied with different levels of effort (e.g. just writing the bare minimum process descriptions, or creating elaborate processes to validate requirements). This results in cases where it is unclear whether a capability should be marked as satisfied or not. Further research is therefore needed to handle this issue.

The current maturity level calculation limits the maturity level based on the lowest unsatisfied capability. This stimulates organizations to stick to the order suggested in the CM, because improving the lowest unsatisfied capability is the only way to improve their maturity score. But it could also result in a low maturity score for an organization that has satisfied almost all capabilities except for a low level capability. This results in a maturity level which does not do justice to the true maturity of the organization. Further research is therefore desirable into a more realistic depiction of maturity than presenting the maturity as simply the lowest score among the focus areas. Accommodating this type of maturity flexibility could also generate more goodwill among organizations being assessed while not doing harm to the truth.

References

Abramovici, M., & Soeg, O. (2002). *Status and Development Trends of Product Lifecycle Management Systems.* Germany: Ruhr-University Bochum, Chair of IT in Mechanical Engineering (ITM).

Appel, W., 2000. *Architecture Capability Assessment*. In Enterprise Planning & Architecture Strategies, vol.4, nr.7, METAGroup.

Bekkers, W. (2008). *Situational Process Improvement in Software Product Management.* Utrecht, The Netherlands: Universiteit Utrecht.

Bekkers, W., Weerd, I. van de, Brinkkemper, S., Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study.*Proceedings of the 21th International Workshop on Product Management,*Barcelona, Spain*,* 41-48.

Brodman, J. G., & Johnson, D. L. (1994). What Small Businesses and Small Organizations Say About the CMM. *Proceedings of the 16th international conference on Software engineering* (pp. 331 - 340). Los Alamitos, CA, USA: IEEE Computer Society Press.

Clements, P., & Northrop, L. (2001). *Software Product Lines: Patterns and Practice.* Reading, MA: Addison Wesley, Boston, MA.

CMMI Product Team. (2002). *Capability Maturity Model Integration (CMMI), Version 1.1, CMU/SEI-2002-TR-012.* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Cusamo. (2004). *The Business of Software.* New York: Free Press.

Dahlstedt, A., & Persson, A. (2003). Requirements Interdependencies - Moulding the State of Research into a Research Agenda. *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2003), held in conjunction with CAiSE 2003*, (pp. 71-80). Austria.

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

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly , 28*, 75-105.

ISO/IEC-15504. (1998). *Information Technology - Software Process Assessment. Technical Report - Type 2.*

Koomen, T., & Baarda, R. (2005). *TMap Test Topics.* Nederland: Tutein Nolthenius.

Kuilboer, J. P., & Ashrafi, N. (2000). Software process and product improvement: An empirical assessment. *Information and Software Technology* , 27-34.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). *Capability Maturity Model for Software (Version 1.1) (SEI/CMU-93-TR-24, ADA263403).* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Reifer, D. J. (2000). The CMMI: it's formidable. *Journal of Systems and Software* , 97-98.

Steenbergen, M., & Brinkkemper, S. (2007). An instrument for the Development of the Enterprise Architecture Practice. *Proceedings of the 9th International Conference on Enterprise Information Systems*, (pp. 14-22).

Vaishnavi, V., & Kuechler, B. (2007, August 4). *Design Research in Information Systems*. Retrieved December 7, 2007, from AISWorld Net: http://www.isworld.org/Researchdesign/drisISworld.htm

Weerd, I. v., Brinkkemper, S., & Versendaal, J. (2007). Concepts for Incremental Method Evolution: Empirical Exploration and validation in Requirements Management. *Proceedings of the 19th International Conference on Advanced Information Systems Engineering* (pp. 469-484). Berlin Heidelberg: Verlag.

Weerd, I. v., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). Towards a Reference Framework for Software Product Mangement. *Proc. of the the 14th Int. Requirements Engineering Conference*, (pp. 312-315). Minneapolis/St. Paul, Minnesota, USA.

Westbrock, T., 2004. *Architecture Process Maturity Revisited and Revised*, METAgroup Delta 2902.

Zahran, S. (1997). *Software Process Improvement: Practical Guidelines for Business Succes.* Reading, MA: Addison-Wesley.