The Situational Assessment Method Put to the Test

Improvements Based on Case Studies

Willem Bekkers, Marco Spruit
Department of Information and Computing Sciences
Utrecht University
Utrecht, the Netherlands
{bekkers, m.r.spruit}@cs.uu.nl

Abstract—**This paper presents an improved Situational Assessment Method (SAM) for Software Product Management (SPM). The improvement is the result of an evaluation process, of which one of the cases is included in this paper. The SAM is tool which allows product managers to evaluate and improve their processes in an incremental manner. It does this by determining which capabilities the organization has currently achieved, and what capabilities should be achieved. A gap analysis can then be performed which results in an advice to implement specific capabilities. This advice includes suggestions for method fragments (best practices) which the organization can use to achieve the capabilities.**

Software Product Management, Software Process Improvement, Maturity Matrix, Competence Model, Situational Assessment Method

I. Introduction

Recent research confirms that Software Product Management (SPM) is a key area within many software companies [1] [2]. A product manager can also be referred to as the "mini-CEO" of an organization [3]. They are positioned at the center of the organization where they keep in contact with all stakeholders to ensure that all stakeholders work towards the same goal according to the strategies set out. As such, a large array of skills is expected; ranging from gathering requirements, to constructing roadmaps.

Although the product manager's function is essential in the product software industry, little education exists in this area [4]. To make things worse, no extensive body of knowledge exists, such as PMBOK [5] and SWEBOK [6] in the fields of project management and software engineering, respectively. This leads to a situation in which product managers learn their skills 'on-the-job', often coming from a position in development, sales, or project management. Problems arise when companies want to professionalize their product management practices, either to support the company's growth, or to make a shift from selling customized software to selling standard product software [7]. Due to the lack of knowledge, increasing the quality of the product by improving the SPM processes is often difficult.

To aid product managers in improving their SPM practices, the SPM Competence Model [4] and the Situational Assessment Method (SAM) [8] were proposed. A key component of the SAM is the SPM Maturity Matrix [9] [10], which is used to determine an organization's SPM maturity level and identify the areas that need improvement to reach a higher maturity level. All kinds of organizations, including small and medium sized organizations, should be able to use the SAM as a guide for incremental process improvement in SPM.

This research focuses on an incremental, or evolutionary, SPI approach for several reasons: a) it is a fundamental way to reduce risk in complex improvement projects [9], and b) this is the natural way for method evolution in many organizations [12] [13].

Since the proposal of SAM, several case studies have been performed which have resulted in the current version of the SAM. This paper presents one of the case studies and the improved SAM. The SAM is currently in an evaluate and improve process.

The structure of the paper is as follows. First, the research approach is explained. After this, the previous work on which the SAM is based and the previous version of the SAM are presented to give some more insight into the workings of the SAM. After this, an example case study and the improved SAM are presented.

II. Research Approach

A. Design research

This study follows the design science methodology, in which research is done through the processes of building and evaluating artifacts [14]. The main artifact in this research is the SAM. The study is performed as action research, since the lead author is both working as a researcher at Utrecht University as well as a consultant at the Dutch product software company Centric.

During our research we follow the five process steps of the design cycle [15]. This design cycle consists of steps that follow an iterative process. Knowledge is produced during the process by constructing and evaluating the artifact, and is used as input for a better awareness of the problem. The five steps are:

1) Awareness of the problem – Section I already laid out the problem under investigation and its context.

2) Suggestion – The suggestion for a solution to the problem identified in step 1 is developed. In this section,

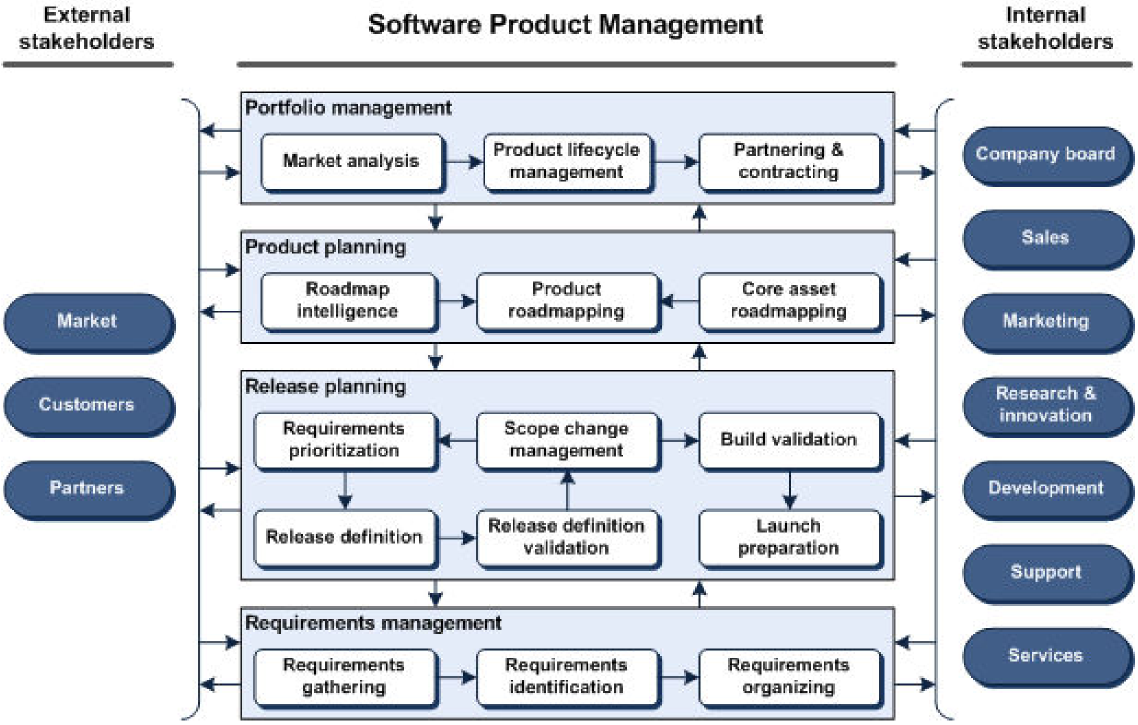

Figure 1. The Software Product Management Competence Model

The approach in tackling the problem and the research methods that are used are described.

3) Development – The artifact that is developed is the SAM for SPM, which is presented in section V.

4) Evaluation – This step comprises the evaluation of the method. We used expert validations, a survey, case studies, and questionnaires to validate the method. The results of these extensive validations lead to a higher level of problem awareness and suggestions for solutions. One of the case studies is presented in section IV.

5) Conclusion – Finally, in section VI, 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. The literature study was based on a multitude of papers describing processes and software process improvement methods within the field of SPM (e.g. [16] & [17]). The literature was used as a starting point for the second part, a brainstorm session. This session was conducted with experts from the scientific community to create a first version of the SAM. Furthermore, an expert validation was held where business professionals validated the results of the brainstorm session. Finally, we performed case studies, to improve the SAM.

The SAM was analyzed in five case studies at product software organizations in the Netherlands to test the applicability in day-to-day business environments. The case studies consisted of the application of the SAM at the organizations and an evaluation on how the organizations looked at the results. This evaluation is used to improve the SAM.

B. *Validity*

To ensure the validity during the research applied multiple triangulation in the following way:

1) Multiple observers – At least one addiontional person will be interviewed for each product manager that is interviewed to verify the data from the first interview.

2) Documentation – All documents used by the product manager during the SPM are requested to verify and clarify data gathered during the interview. Example documents include requirement templates, product roadmaps, and release documents.

To guarantee the reliability of the case study, a chain of evidence is created recording all procedures. This is done by using a case study protocol and by maintaining a case study database storing all relevant information used in the case study.

III. Previous Work

This section describes the six components which constitute the Situational Assessment Method: (A) the SPM Competence Model, (B) Capabilities, (C) the SPM Maturity Matrix (D) Situational Factors and Effects (E) method fragments (F) the Situational Assessment Method.

A. SPM Competence Model

The SPM Competence Model [9] (Figure 1) presents an overview of all of the areas which are important to the field of the SPM. These areas are called focus areas. The relevant external and internal stakeholders are presented on the left and right sides of the model. The model does not include the development departments' activities of the product software organization. Development is simply one of the stakeholders that provides input for the SPM processes.

Four main business functions are defined in the model, namely: Requirements management, Release planning, Product planning, and Portfolio management. These business functions are based on the structure where a portfolio consists of products, a product consists of releases, and releases consist of requirements. The portfolio is represented in the Portfolio management function. The products are represented in the Product planning function. The releases are represented in the Release planning function. And finally, the requirements are represented in the Requirements management function.

Each business function consists of a number of focus areas (the white areas in figure 1), each of which represents a strongly coherent group of capabilities within a business function. The SPM Competence Model consists of 15 focus areas which are explained in more detail in [9].

B. Capabilities

We define a capability as "a predefined goal that needs to be achieved to reach the maturity level with which it is associated" in our research. The SPM Maturity Matrix contains 68 capabilities. The number of capabilities per focus area ranges from three to six. We describe the following attributes for the capabilities in the SPM Maturity Matrix:

1) Name – A name shortly describing action required by the capability.

2) Goal – The goal describes the purpose of the capability, It indicates the advantage of executing the capability.

3) Action – The action describes what must be done in order to meet the capability.

4) 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.

5) 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.

Table I shows a capability from the SAM for SPM.

TABLE I. THE REQUIREMENT DEPENDENCY LINKING CAPABILITY FROM THE SOFTWARE PRODUCT MANAGEMENT MATURITY MATRIX [9]

Name	Requirement dependency linking
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)	Requirements gathering A
Reference(s)	Dahlstedt & Persson (2003)

C. SPM Maturity Matrix

The SPM Maturity Matrix is a key component of the SAM for SPM. It is structured based on the SPM Competence Model and presents all of the important practices – the capabilities – in a best practice order for implementation, so that organizations have a guideline for the improvement of those SPM practices. Organizations can thus identify areas of improvement by comparing their organization's processes to the capabilities in the SPM Maturity Matrix. Based on the best practice order provided by the maturity matrix, organizations can plan the improvement of their processes.

The maturity matrix depicted in Table II is a Focus Area Maturity Model [20] [21]. We chose to develop this type of maturity model because of the shortcomings of other existing models described in [22] and to enable local analysis and incremental improvement. Focus area maturity models have already successfully been developed in the testing domain [23] and the architecture domain [21].

A focus area maturity model consists of a number of focus areas, each with its own number of specific maturity levels. The focus areas are represented in the leftmost column in Table II. The focus area specific maturity levels (capabilities) are represented by the letters A-F in Table II and range from maturity level 1 to 10 (the topmost row in Table II). Their spread across the overall maturity levels indicates a best practice order, in which capabilities in the maturity matrix are implemented from left to right.

The capabilities have been carefully balanced throughout the different maturity levels to ensure that the amount of work needed to increase the maturity level to the next is in balanced logically over all the maturity levels.

The development steps of a Focus Area Maturity Matrix in general, and the maturity matrix in particular are discussed in length in [20].

TABLE II. FOUR MAIN BUSINESS FUNCTIONS ARE DEFINED IN THE MODEL, NAMELY: REQUIREMENTS MANAGEMENT, RELEASE PLANNING, THE SOFTWARE PRODUCT MANAGEMENT MATURITY MATRIX [9]

	0	1	2	3	4	5	6	7	8	9	10
Requirements management											
Requirements gathering		A		B	C		D	E	F		
Requirements identification			A			B		C			D
Requirements organizing				A		B		C			
Release planning											
Requirements prioritization			A		B	C	D			E	
Release definition			A	B	C				D		E
Release definition validation					A			B		C	
Scope change management				A		B		C		D	
Build validation					A			B		C	
Launch preparation		A		B		C	D		E		F
Product planning											
Roadmap intelligence				A		B	C		D	E	
Core asset roadmapping					A		B		C		D
Product roadmapping			A	B			C	D		E	
Portfolio management											
Market analysis					A		B	C	D		E
Partnering & contracting						A	B		C	D	E
Product lifecycle management					A	B			C	D	E

D. *Situational Factors and Effects*

A Situational Factor (SF) 'contains information about the process, the context of the organization, and the organization itself' [24]. SFs describe the situational context in which 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 [25]. 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.

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.

Four examples of SFEs from the SAM for SPM can be seen in Table III. 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.

TABLE III. THE REQUIREMENT DEPENDENCY LINKING CAPABILITY FROM THE SOFTWARE PRODUCT MANAGEMENT MATURITY MATRIX

Situational Factor	**Operator**	**Value**	**Capability**	**Effect**
Customer involvement	=	Low	Requirements gathering F	Disable
Customer variability	>	10%	Requirements Identification C	Enable
Partner involvement	=	Low	Requirements gathering F	Disable
Partner involvement	=	Low	Requirements prioritization C	Disable

E. *Method Fragments*

In this research, method engineering is used to analyze and store information on method fragments [18], consisting of processes and their deliverables, in product software companies. To achieve this method administration, Process Deliverable Diagrams [19] will be used to model activities and work products. These diagrams are a way of meta-modeling a process by creating a model with formal descriptions.

F. *The Situational Assessment Method*

The SAM is a conceptually generic assessment method with domain-specific implementations. The variant described in this paper focuses on software product management. It is a Situational Assessment Method for Software Product

Management (SAM for SPM). The goal of the SAM is to identify improvement areas within the SPM organization.

The SAM 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 [20] [21]. 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.

There are five important components in the SAM: a knowledge base and three process steps (questionnaire, calculation, and feedback).

1) Knowledge base – The knowledge base contains the knowledge on which the assessment advice is based: the SPM Maturity Matrix; the Situational Factors for SPM; the Situational Factor Effects (the effects certain specific SF values have on the capabilities in the maturity matrix); and the method fragments.

2) 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 Situational Factor values for the organization.

3) 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, the optimal maturity is modeled in the Optimal Capability Profile. 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.

4) 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 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 the improvements in increments, or all at once. This allows the organization to determine how much resources 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 also 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 business function. This allows the organization to assess a specific aspect of its organization (e.g. Requirements gathering, or Portfolio management).

IV. Case Study

A. An Example Case Study

The case study presented in this section is one of the case studies we performed during the design research process. It is meant to serve as an example and provides a case how the SAM can be applied in practice. The Situational Factor values for this case have been gathered but are not included in this paper due to confidentiality constraints.

The organization at which the case study of the SAM was performed is active in the governmental sector. They have a great need for reliable roadmaps since the government requires higher standards of SPM organizations than the average organization. They therefore indicated that they wanted to use the SAM to investigate the Product planning and Portfolio management business functions.

Firstly, Table IV shows the capabilities which were already achieved by the organization in question (the Current Capabilities Profile). The cells shaded green show the achieved capabilities and the progression through the maturity matrix. The organization was at maturity level 2 overall. Or level 2 for Product planning and 3 for Portfolio management if one opts to assess the business functions separately. Capability B of Product roadmapping has not been achieved by the organization, but capabilities C and D have been. This is possible since the maturity matrix gives a best practice order for the capabilities. It is possible that organizations have implemented capabilities in a different order if the higher placed capabilities have no prerequisite pointing to the skipped capability.

Secondly, Table V presents the optimal capability profile. The cells shaded red indicate the capabilities that have been disabled. Only one capability was thus disabled (capability E of Product Lifecycle Management) since the organization being assessed was quite large and the product will remain in production for many years.

Thirdly, Table VI (the Areas of Improvement Matrix) gives an overview of the capabilities that the organization still needs to achieve to increase their maturity to the highest level (the capabilities indicated as 'missing').

Finally, Table VII shows the Current Capability Profile that will be the result of the implementation of the capabilities that were selected by the organization to be included in the first incremental improvement (the cells shaded green). This increment will, once completed, result in a Current Capability Profile with a maturity level of 7.

The organization indicated that the SAM, the SPM Competence Model, and the SPM Maturity Matrix provided them with a much needed structure for their SPM organization. They indicated that they already gathered some of the information, but often ended up not using and eventually losing the information and reinventing the wheel (processes and templates) every few years because they had not structured their processes. Other improve points were an increase in knowledge about their competitors. And a better balance between the different products in their portfolio. This was the result of an increase in knowledge sharing about the turnover and profit of the different products. And also because of a better high level overview over all of the products, which was made possible because of the comparable and substantiated roadmaps.

The organization did initially struggle with the implementation of the capabilities as they did not know how to define the processes. They wanted some method fragments for them to start with.

The RASCI model was used during the planning of the adoption of the new capabilities. Which proved to be a valuable tool to map the responsibilities of the various stakeholders for the various capabilities.

TABLE IV. THE CURRENT CAPABILITIES PROFILE

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Product planning											
Roadmap intelligence	🟩	🟩	🟩	A	🟩	B	C		D	E	
Core asset roadmapping	🟩	🟩	🟩	🟩	A	🟩	B		C		D
Product roadmapping	🟩	🟩	A	B			C	D		E	
Portfolio management											
Market analysis	🟩	🟩	🟩	🟩	A	🟩	B	C	D		E
Partnering & contracting	🟩	🟩	🟩	🟩	🟩	A	B	🟩	C	D	E
Product lifecycle management	🟩	🟩	🟩	🟩	A	B			C	D	E

TABLE V. THE OPTIMAL CAPABILITIES PROFILE

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Product planning											
Roadmap intelligence				A		B	C		D	E	
Core asset roadmapping					A		B		C		D
Product roadmapping			A	B			C	D		E	
Portfolio management											
Market analysis					A		B	C	D		E
Partnering & contracting						A	B		C	D	E
Product lifecycle management					A	B			C	D	E

TABLE VI. THE AREAS OF IMPROVEMENT MATRIX

	A	**B**	**C**	**D**	**E**
Product planning					
Roadmap intelligence	Implemented	Missing	Missing	Missing	Missing
Core asset roadmapping	Implemented	Missing	Missing	Missing	
Product roadmapping	Implemented	Missing	Implemented	Implemented	
Portfolio management					
Market analysis	Implemented	Implemented	Missing	Missing	Missing
Partnering & contracting	Implemented	Implemented	Implemented	Missing	Missing
Product lifecycle management	Missing	Missing	Missing	Missing	-

TABLE VII. THE CHOSEN IMPROVEMENT INCREMENT

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Product planning											
Roadmap intelligence	🟩	🟩	🟩	A	🟩	B	C	🟩	D	E	🟩
Core asset roadmapping	🟩	🟩	🟩	🟩	A	🟩	B	🟩	C		D
Product roadmapping	🟩	🟩	A	B	🟩	🟩	C	D	🟩	E	🟩
Portfolio management											
Market analysis	🟩	🟩	🟩	🟩	A	🟩	B	C	D	🟩	E
Partnering & contracting	🟩	🟩	🟩	🟩	🟩	A	B	🟩	C	D	E
Product lifecycle management	🟩	🟩	🟩	🟩	A	B	🟩	🟩	C	D	

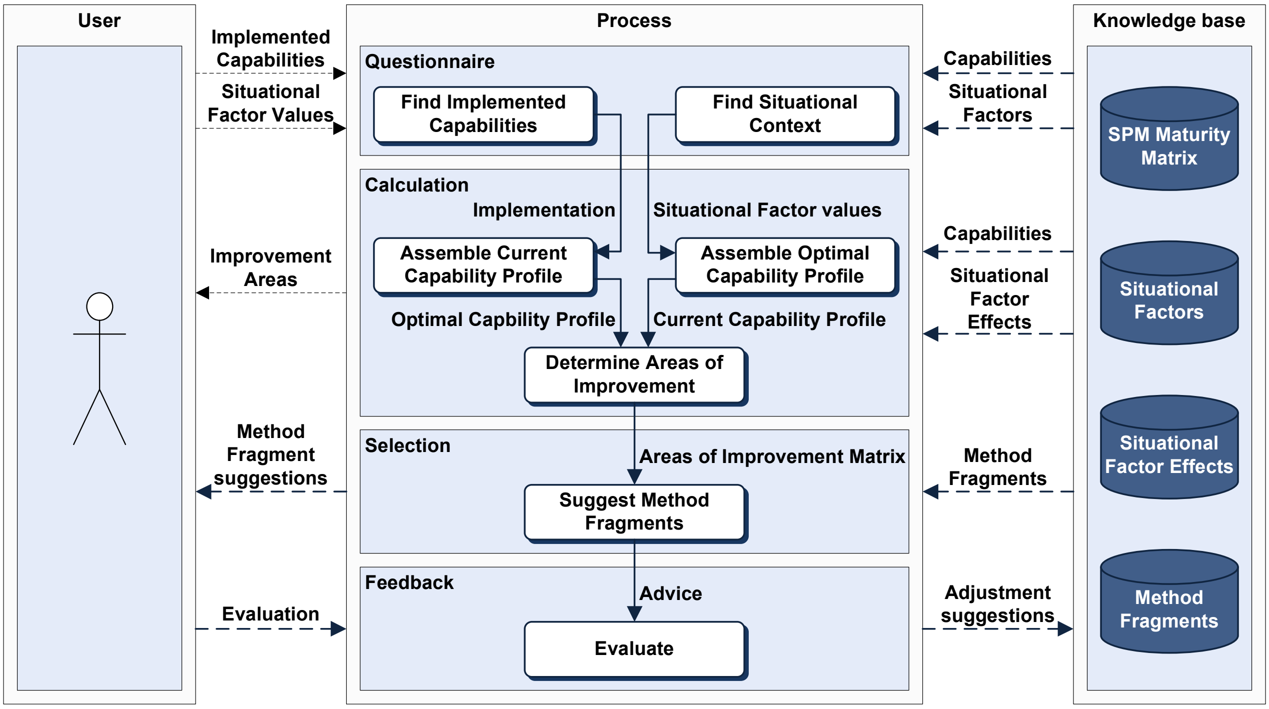
User
Process
Knowledge base
Implemented Capabilities
Situational Factor Values
Improvement Areas
Method Fragment suggestions
Evaluation
Questionnaire
Find Implemented Capabilities
Find Situational Context
Calculation
Implementation
Situational Factor values
Assemble Current Capability Profile
Assemble Optimal Capability Profile
Optimal Capbility Profile
Current Capability Profile
Determine Areas of Improvement
Selection
Areas of Improvement Matrix
Suggest Method Fragments
Feedback
Advice
Evaluate
Capabilities
Situational Factors
Capabilities
Situational Factor Effects
Method Fragments
Adjustment suggestions
SPM Maturity Matrix
Situational Factors
Situational Factor Effects
Method Fragments

Figure 2. The Situational Assessment Method

B. *General Conclusions of the Cases*

All of the organizations at which the cases were performed were positive about the advice they received from the SAM. They indicated that they really appreciated the tangible nature of the capabilities. The capabilities were not too abstract or vague to be able to use them. They also indicated that they valued the fact that the SAM does not require them to do perform capabilities that are not useful for their organizations.

The questionnaires also proved to be understandable to the users. The questions were not too abstract, and did not use language which was unknown to the users. One thing the users did have some difficulties with was the yes/no answer form. Some users indicated that would like to have a less strict form. They indicated that there were some capabilities which they often performed, but not all of the time. They now had to answer these questions with no, where they would like to answer them with 'in most cases'.

A point that all organizations struggled with was how to tackle these capabilities. They had no idea where to start the search for method fragments. And those that did search did not have the time to perform a decent search. This lack of knowledge and time resulted in the organizations starting to crudely reinvent the wheel over and over. They designed new processes to tackle the capabilities without looking at existing methods or even looking at neighboring organizations who already perform these capabilities. They all indicated a wish to be guided in this method selection process as well.

V. The Situational Assessment Method

The results of the case studies showed that the SAM works. But it also became apparent that there was a great wish from the organizations to take the SAM one step further and also suggest method fragments which they can use with the specific capabilities. This section presents an improved version of the SAM which includes a process step in which method fragments are selected based on the SFs that describe the organization being assessed (see figure 2).

The new components within the SAM are the method fragments in the knowledge base, and a new process step, called selection, which selects appropriate method fragments for the organization.

The dotted arrows in Figure 2 indicate the flow of information between the user and the assessment process, and between the assessment process and the knowledge base. The arrows within the process show the process flow in the assessment.

A. *Knowledge base*

The knowledge base contains all of the information needed to perform the assessment and provide an advice. One of the components in the Knowledge base is the SPM Maturity Matrix (see Table II). This maturity matrix provides an overview of all the capabilities that need to be achieved to reach a full-grown maturity. The maturity 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 10 represent the maturity levels (where 0 is low and 10 is high). 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 three (A-C) to six (A-F). The maturity matrix also 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 [9].

The capabilities in the SPM Maturity Matrix are processes, the use of standards, and technical instruments that need to be achieved by SPM organizations 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.

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 for 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 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).

The second component of the Knowledge base consists of the Situational Factors. The SFs are used to model the context in which the organization being assessed operates.

The effects the SFs have on the SPM Maturity Matrix are modelled using the Situational Factor Effects, which indicate which capabilities in the SPM Maturity Matrix apply in the context of the organization. Capabilities can be disabled (if they are enabled by default) or disabled (if they are enabled by default.

The current SFEs in the Knowledge base are based on the knowledge of experts. This knowledge was gathered during expert interviews with experts from the scientific community and with experts from practice.

B. 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 for SPM (see Table VIII for two examples from the questionnaire). The second questionnaire is the 'Implemented capabilities' questionnaire. This questionnaire determines which capabilities are implemented within the organization. It consists of one statement for each of the capabilities in the SPM Maturity Matrix, 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 68 questions in this questionnaire (see Table IX 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 this section) allow for the creation of an intelligent 'Implemented capabilities' 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. We can also use the SF values to reduce the number of questions in this questionaire. 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 and first performing the Situational context questionnaire.

TABLE VIII. TWO EXAMPLES FROM THE ‘SITUATIONAL CONTEXT’ QUESTIONNAIRE IN THE SAM FOR SPM

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 IX. TWO EXAMPLES FROM THE ‘IMPLEMENTED CAPABILITIES’ QUESTIONNAIRE IN THE SAM FOR SPM

Code	Statement	Answer
Requirements identification B	The correctness (“Is the definition correct?“) and completeness (“Does the requirement describe all relevant aspects?”) of the requirement is validated.	No
Scope change management C	An impact analysis is performed to determine the effects of the scope change.	Yes

C. 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.

Firstly, the Current Capability Profile 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, a custom maturity matrix tailored to the situational context of the organization. Note that some of the capabilities of the maturity matrix may be disabled in this tailored version, because they are not relevant in situational context of the organization.

Finally, the improvement areas are determined with the help of a gap analysis. These are determined by comparing the Current Capability Profile with the Optimal Capability Profile. 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). This matrix 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 presented [20], 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 maturity matrix 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 Requirements gathering A and Launch preparation A are satisfied, and e.g. Requirements identification A has not been satisfied, then the maturity level is 1.

D. *Selection*

The selection stage selects method fragments that fit the organization and suggests these to the organization.

The method fragments are stored in the Knowledge Base in the form of PDDs (see section III A). They are organized according to two aspects:

1) Capability linking – To be useful the method fragment should cover at least one of the capabilities in the SPM Maturity Matrix. The method fragments should therefore be linked to at least one capability. A more elaborate method fragment can also cover multiple capabilities. There are, for example, multiple capabilities that describe parties that should be involved in the prioritization process. These could be covered by one method fragment that elaborately covers the prioritization process. The method fragment would then be linked to all of these capabilities.

2) Situational Factor restrictions – Method fragments are usually not suited for all types of organizations. The useful situations can be expressed using the SFs. For example, a prioritization process in which all customers are visited by a sales representive of the orgazation will only be realistic when an organization only has a select number of customers. In this case the SF 'Number of customers' could be used to indicate a range of customers with which this method can be applied efficiently.

At this point in the process the capabilities which need improvement are know. The SAM can now select the best suited method fragments by looking at the method fragments which cover those capabilities, and comparing the results from the Situational Context questionnaire with the SF restrictions

E. *Feedback*

The field of SPM is constantly evolving. The content of the knowledge base will therefore need to evolve with the SPM field.

There are two parts to the evaluation. Firstly, the method fragments are rated. The organization can indicate how useful they find the method fragment, how hard they found the implementation, etc. This rating, in combination with the situational context of the organization, allows for better method fragment suggestions in future assessments because we can now better determine which methods organizations with similar profiles find useful.

Secondy, the evaluation provides feedback which is used 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.

The current SPM Maturity Matrix and Situational Factors are fully developed. The Situational Factor Effects and method fragments are in an early stage of development. The Knowledge Base components are currently based on expert interviews (both from the scientific and practical fields), case studies, and scientific literature.

VI. Preliminary Conclusions & Future research

A. *Preliminary Conclusions*

We believe that the improved Situational Assessment Method provides organizations with an elaborate tool for self assessment and improvement. The organizations all found the SAM very useful as can be read in the case study and the general conclusions of the case studies. They also all indicated a wish to take the model one step further and also provide them with methods which they can use. The SAM now fulfills this wish with the inclusion of the selection step and the method fragments.

The combination of method fragments with SFs also offers new possibilities. We can now link organizational profiles to capabilities which can provide insight into who uses specific method fragments. More insight into which methods actually work in practice can also be achieved because the organization rate the method fragments.

B. *Future Research*

The SAM now almost fully covers the Product Software Knowledge Infrastructure (PSKI) as suggested in [26]. The PSKI also links method fragments together automatically to form one large process. Further research towards the merging of the method fragments is useful since it makes the advice from the process even simpler to pick up for organizations.

Further research into the Situational Factor Effects is needed. There are currently relatively few effects modelled in the Knowledge Base. This provides some fine-tuning, but the level of fine-tuning should be increased to provide the full benefit of the SAM.

The collection of method fragments in the Knowledge Base should be expanded to provide a wider range of choices allowing for the selection of even better fitting method fragments.

The implemented capabilities are currently determined using strict yes/no questions. In reality we often came across the situation 'yes, but' or 'most of the time'. It would be useful to research alternative mechanisms for determining the implemented capabilities.

ACKNOWLEDGMENT

Gratitude goes out to Centric IT Solutions for their funding which allow the author to perform this research.

REFERENCES

[1] Fricker, S., Gorschek, T., Byman, C., & Schmidle (2010), "A. Handshaking: Negotiate to Provoke the Right Understanding of Requirements". IEEE Software 26(6).

[2] P. Berander, "Evolving Prioritization for Software Product Management," in APS, PhD thesis, Ronneby: Bleking Institute of Technology, 2007

[3] Christof Ebert, Sjaak Brinkkemper, Slinger Jansen, Gerald Heller, "2nd International Workshop on Software Product Management," International Workshop on Software Product Management, pp. i-ii, 2008 Second International Workshop on Software Product Management.

[4] Weerd, I. van de, Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). "Towards a reference framework for software product management". Proceedings of the 14th International Requirements Engineering Conference, Minneapolis/St. Paul, Minnesota, USA, 319-322.

[5] Project Management Institute. "A guide to the project management body of knowledge" (PMBOKGuide), 2000 ed. Newtown Square, PA: Project Management Institute; 2000.

[6] Abran, A., Moore, J. W., Bourque, P., Dupuis, R., & Tripp, L. L. (2004). "SWEBOK: Guide to the Software Engineering Body of Knowledge". Los Alamitos, CA: IEEE Computer Society.

[7] Artz, P., Weerd, I. van de, & Brinkkemper, S. (2010). "Productization: transforming from developing customer-specific software to product software". Proceedings of the 1st International Conference on Software Business (ICSOB 2010), LNBIP 51, 90–102.

[8] Bekkers, W., Spruit, M., Weerd, I. v., & Brinkkemper, S. (2010). "A Situational Assessment Method for Software Product Management". Proceedings of the 18th European Conference on Information Systems (ECIS2010), June 7-9, Pretoria, South Africa

[9] Bekkers, W., Weerd, I. van de, Spruit, M., and Brinkkemper, S. (2010). "A framework for process improvement in software product management". Accepted for the 17th EuroSPI conference, September 1-3, Grenoble, France.

[10] Bekkers, W., Weerd, I. van de (2010). "SPM Maturity Matrix" , Technical Report UU-CS-2010-013, Utrecht, The Netherlands: Utrecht University, 2010.

[11] Krzanik, L., & Simila, J. (1997). "Is my software process improvement suitable for incremental deployment?". Proceedings of the 8th International Workshop on Software Technology and Engineering Practice. London, UK, 76-87.

[12] Weerd, I. van de, Brinkkemper, S., & Versendaal, J. (2010). "Incremental method evolution in global software product management: A retrospective case study". Accepted for publication in the Journal of Information & Software Technology.

[13] Weerd, I. van de, Versendaal, J., & Brinkkemper, S. (2006). "A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management". Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06), Luxembourg, 97-112.

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

[15] Vaishnavi, V., Kuechler, B. (2009, August 01). Design Research in Information Systems. Retrieved Mach 16, 2010, from AISWorld Net: http://www.isworld.org/Researchdesign/drisISworld.htm

[16] 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).

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

[18] Brinkkemper, S. (1996). "Method engineering: engineering of information systems development methods and tools". Information and Software Technology , 38 (4), 275-280.

[19] Weerd, I. v., Souer, J., Versendaal, J., & Brinkkemper, S. (2005). "Situational Requirements Engineering of Web Content Management Implementations". International Workshop on Situational Requirements Engineering Processes: Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes (pp. 13-30). Paris: University of Limerick.

[20] Steenbergen, M., Bos, R., Brinkkemper, S., Weerd, van, I., Bekkers, W. (2010). "The Design of Focus Area Maturity Models. Proceedings of the 5th International Conference on Design" Science Research in Information Systems and Technology (DESRIST2010).

[21] 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).

[22] Bekkers, W., Weerd, I. van de, Brinkkemper, S., Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study. Presented at the 21th International Workshop on Software Product Management (IWSPM2008), Barcelona, Spain, September 9, 2008.

[23] Koomen, T., & Baarda, R. (2005). "TMap Test Topics". Nederland: Tutein Nolthenius.

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

[25] 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.

[26] Weerd, I. v., Versendaal, J., & Brinkkemper, S. (2006). A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management. Twelfth Working Conference on Requirements Engineering: Foundation for Software Quality, (pp. 97-112). Luxembourgh.