Journal of Universal Computer Science, vol. 30, no. 5 (2024), 645-673
submitted: 9/8/2023, accepted: 11/12/2023, appeared: 28/5/2024

Software Process Improvement by Managing Situational Method Engineering Knowledge

Razieh Dehghani
(Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
https://orcid.org/0000-0003-4619-0914, razieh.dehghani@sharif.edu)

Raman Ramsin
(Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
https://orcid.org/0000-0003-1996-9906, ramsin@sharif.edu)

Abstract: Organizational processes have been recognized as valuable knowledge assets. Situational Method Engineering (SME) processes are particularly valuable as they are used for engineering other processes: SME processes help construct bespoke Software Development Methodologies (SDMs) for specific software-engineering project situations. Every SDM has a Software Development Process (SDP) at its heart, which specifies the activities that should be performed throughout the project, the products that should be produced, and the people that should be involved. Existing SME methods suffer from certain weaknesses that are rooted in loss of knowledge within their processes; for instance, the method engineers' experience, which is a kind of tacit knowledge, is not properly captured and utilized in these processes. Managing SME process knowledge helps alleviate these weaknesses through reusing the software developers' experience and maintaining the method engineers' knowledge. We propose an evaluation framework that can be used for assessing an SME method's ability to manage process knowledge. We also provide a model that guides the improvement of existing SME methods in their support for Knowledge Management (KM), and also helps engineer new SME methods that provide adequate KM support. We have assessed the applicability of the proposed evaluation framework and improvement model by using them to enhance eight prominent SME methods, and also by applying them to four industrial case studies.

Keywords: Software Process Knowledge, Software Process Improvement, Knowledge Management, Software Development Methodology, Situational Method Engineering
Categories: D.2.1, D.2.2, D.2.9, M
DOI: 10.3897/jucs.110894

1 Introduction

A Software Development Methodology (SDM) is a framework for applying software development practices in a disciplined manner with the aim of engineering software products. A SDM (commonly referred to simply as "methodology") is made up of two parts: a Software Development Process (SDP) and a Modeling Language (ML); the SDP part consists of three components: 1) the products that should be produced (including the target software system), 2) the activities required to develop the products, and 3) the skills required for performing the activities [Ramsin and Paige, 2008].

The diversity of software applications has resulted in the need for a range of SDMs, but available SDMs do not satisfy the diverse requirements of software developers [Henderson-Sellers et al., 2014]. Situational Method Engineering (SME) provides a solution to this problem through constructing bespoke SDMs for specific project situations. Although SME has been described as a "social process that needs to pay attention to human factors such as values, attitudes, and knowledge" [Henderson-Sellers et al., 2014], SME processes have not been researched as special kinds of knowledge-intensive processes.

There are three levels of abstraction for putting SDPs into use: 1) at the lowest level, software developers use a specific customized SDP for engineering software, 2) at a higher level, project managers analyze the characteristics of the software development project at hand and customize an available ML and SDP, and 3) finally, at the highest level, method engineers follow a SME method and use their knowledge to analyze the characteristics of the situation, and then propose the most appropriate ML and construct a practical, tailored-to-fit SDP.

Typically, method engineers work in isolation and need to acquire knowledge from developers and project managers. Thus, the problem that this work aims to address is the lack of attention to the need for reusing the experience previously acquired by individuals who have used or engineered SDMs. Experience is a kind of tacit knowledge that is particularly difficult to capture and reuse [Alavi and Leidner, 2001], and from a Knowledge Management (KM) standpoint, available SME methods suffer from serious weaknesses, especially in reusing developers' experience in operationalizing SDMs. In general, these weaknesses might result in the following knowledge risks (adapted from [Durst and Zieba, 2019]):

- "Knowledge attrition": The SDPs or the products that are produced throughout the SDP become obsolete. Besides, they might be used inappropriately.
- "Knowledge loss": The knowledge of individuals (software developers/method engineers/project managers), or their products, are lost.
- "Knowledge leakage": Right of ownership is violated for products and SDPs.
- "Knowledge spillover": Those products, people, or SDPs that create competitive advantage are not preserved.
- "Lost reputation": The SDM seems impractical due to negative points of view about the skills of individuals, quality of products, and practicality of SDPs.
- "Lost sustainability": The SDM is not flexible enough, and it is difficult to configure it to satisfy the new development requirements.

To avoid the above-mentioned risks, two main questions need to be answered: 1) Which metrics can be used to evaluate SME methods (and thus SDPs) as to their capability to preserve and share SME (and thus software development) knowledge? And 2) Are KM practices appropriate solutions to improve SME methods (and thus SDPs)? To find the answers, two main goals have respectively been targeted in this research: 1) proposing an evaluation framework to find the strengths and weaknesses of SME methods, and 2) proposing an improvement model to use appropriate KM practices to alleviate the weaknesses. Our proposed framework encompasses a set of evaluation criteria that have been extracted by exploring the knowledge-intensive features of SME methods and SDMs; also included in the framework is a knowledge flow visualization schema to assess the knowledge flows supported by SME methods. Method engineers and project managers can use the evaluation framework to find the weaknesses of in-use processes, and then apply the improvement model to improve them. To evaluate the proposed evaluation framework and improvement model, we have applied them to eight SME methods, and have also conducted four case studies. It was found that the processes to which the framework and model were applied had indeed been improved in terms of the quality factors important to process stakeholders.

The rest of this paper is structured as follows: Section 2 explains the structure of the research conducted; Section 3 provides a review of the relevant research; Section 4 describes the proposed evaluation framework; Section 5 presents the proposed improvement model; Section 6 provides the results of evaluating the proposed evaluation framework and improvement model; and finally, Section 7 provides the conclusions and suggests directions for furthering this research.

2 Research Structure

This research was conducted in three main steps: developing the evaluation framework, building the improvement model, and empirical validation of the framework and model. Figure 1 shows the main activities, inputs, outputs, and evaluation strategies for each of these steps. According to Hevner et al.'s research guidelines [Hevner et al., 2004], all outputs (main outcomes) produced throughout the research process steps should be evaluated through auxiliary products. Thus, in each step, our evaluation strategy has been based on intermediate assessment of the outputs. For example, the strengths and weaknesses of SME methods identified by the evaluation framework have in turn been used for assessing the proposed framework's capability to reveal distinguishing KM-driven features of SME methods. Even though these three steps were performed sequentially, learning loops were added to address the weaknesses found during the evaluations. The motivations and methodology behind each of the steps are as follows:

1. **Building the evaluation framework:** Since the framework should find deviations from the "Should-Be" (ideal) state of SDMs and SME methods, ideal features have first been elicited. To this aim, considering the low number of research works conducted in the area of managing SME knowledge, relevant SME, SDM, and KM resources were identified and explored through backward and forward snowballing [Jalali and Wohlin, 2012]. By using these resources, knowledge-driven challenges and opportunities have been elicited as "Should-Be" features. In other words, requirements have been identified that should be satisfied to ensure proper elicitation/maintenance of SME knowledge. Thereafter, for each feature, a criterion has been proposed to assess the level to which the feature might be supported; in other words, the requirements are manifest in a set of criteria for evaluating SME methods. Also, to show a high-level schema of an SME method's capability to flow the knowledge appropriately, a multi-dimensional schema has been proposed to visualize the capability to transfer (provide) the right SME knowledge (people, product, process [Henderson-Sellers et al., 2014]), to the right method engineer/project engineer/developer, at the right time [Riege, 2005]. The framework was then analyzed and improved through applying it to a select set of existing SME methods: the results were analyzed so as to assess the proposed framework's capability to reveal the strengths and weaknesses of SME methods, and the shortcomings thus identified were addressed through improving the framework as to comprehensiveness and precision; in addition, a set of meta-criteria were used for assessing and improving the evaluation framework.

2. **Building the improvement model:** This step aims to guide method engineers to share their knowledge so as to prevent rework and also to improve the efficiency of the SME process and its main output, which is a bespoke SDM engineered for the project situation at hand. To this aim, KM practices that could be used for satisfying the KM requirements of SME processes were identified. These practices should help address the problems revealed in the previous step. For this purpose, the improvement structure proposed by CMMI models [Team–CMMI Product, 2010] was used. The improvement model was then developed to provide guidance on how to use suitable KM practice(s) for satisfying KM requirements. To validate the model intermediately, it has been verified that applying it to the targeted SME methods does indeed improve them. The results have shown that the model is capable of establishing (or improving) the knowledge flows required in an SME process.
3. **Final evaluation:** To evaluate the evaluation framework and improvement model empirically, we have followed the guidelines put forward in [Brereton et al., 2008, Kitchenham et al., 2008, Jedlitschka and Pfahl, 2005, Runeson et al., 2012] for conducting case studies. For this purpose, the context and subjects have first been chosen: four companies with differing knowledge risks were selected, and their method engineers, project managers, and developers were targeted as subjects for data collection. Two rounds of data collection were conducted to collect data on the "as-is" and "improved" situations. Next, based on the goals targeted, appropriate quantitative data analysis metrics were chosen, and the data were analysed and reported. It should be noted that we have conducted a pilot round of study, through which experts in SME and SDP have confirmed that the goals of questions were understood clearly (validity). Also, some questions were intentionally aimed at checking the same issue in different ways, and the responses confirmed this issue (reliability).

	Main Activities	Input	Output	Intermediate Evaluation Strategy
1. Building the Evaluation Framework	1) Identifying requirements for managing SME process knowledge; 2) Identifying requirements for establishing appropriate SME knowledge flows	1) Features of Knowledge Intensive Processes; 2) Knowledge-Driven Features of SME Methods; 3) Knowledge-Driven Features of SDMs	1) KM-Based Evaluation Criteria; 2) Knowledge Flow Visualization Method; 3) Strengths and Weaknesses of Available SME Methods	1) Using the framework to evaluate eight prominent SME methods; 2) Using a set of meta-criteria
2. Building the Improvement Model	1) Collecting best KM practices; 2) Linking KM practices to the evaluation criteria	1) Proposed Evaluation Framework; 2) KM Practices; 3) Strengths and Weaknesses of Available SME Methods	1) KM-Driven Model for Improving SME Methods (and thus SDMs); 2) Improved Version of Eight SME Methods (evaluated in previous step)	1) Using the model to improve eight prominent SME methods
3. Final Evaluation	1) Choosing appropriate companies; 2) Collecting data; 3) Analyzing Data; 4) Reporting	1) Proposed Evaluation Framework; 2) Proposed Improvement Model	1) Results of Qualitative Evaluation; 2) Results of Quantitative Evaluation	1) Validating the questions and data

Figure 1: Research Process (generally inspired by [Hevner et al., 2004])

3 Literature Review

Engineering situational SDMs requires "knowledge of what works in what situation, and why" [Henderson-Sellers et al., 2014]. These aspects of SME knowledge have been investigated by researchers separately, even though establishing appropriate SME knowledge flows requires transferring all of these aspects of knowledge to the right person (method engineer/project engineer/software developer) at the right time [Riege, 2005]. The following paragraphs discuss the previous research conducted in this area, along with the need for an evaluation framework.

Clarke and O'Connor have presented a categorization of general situational factors which should be considered throughout the process of building SDMs [Clarke and O'Connor, 2012]. Some special-purpose factors have also been proposed. For example, engineering agile processes requires investigating specific managerial factors [Rasnacis and Berzisa, 2017]. These are valuable knowledge repositories that help analyze the situations, but instantiating the factors requires understanding the strengths and weaknesses of previous/current in-use SDPs. To improve the in-use SDPs, researchers have provided guidelines; CMMI-for-Development is a good example [Team–CMMI Product, 2010]. However, since the rationale behind improvement guidelines is not explained, parts of the improvement knowledge are not transferred; this may result in confusion and inability to tailor the processes in tandem with situational changes.

Software Process Meta-Models (SPMM) help avoid misunderstandings by using a common description language [Kuhrmann et al., 2013]. Unfortunately, available languages do not cover all aspects of SME knowledge; for example, the metaprocess proposed in [Engels and Sauer, 2010] focuses on structural and behavioral dimensions only. Since different languages have been used for describing the outputs of SME methods, a unique comprehensive description language should be proposed and used by all method- and software engineers. In addition to description languages, which provide a top-down approach for transferring SME knowledge, process mining tools have been engineered to analyze the in-use SDPs; these tools provide a bottom-up approach for exploring processes and finding process patterns [Bose et al., 2011].

In conclusion, existing literature has failed to address KM concerns adequately, resulting in the following research gaps: 1) inability to map situational factors to SME practices, which requires preserving and sharing the experience acquired throughout the process of engineering and using SDMs, 2) failure to provide guidance on how to make a situational comparison of available SME methods, so that the most appropriate method can be chosen for the software development project at hand, and 3) lack of adequate mechanisms for preparing a repository of data about in-use SDPs, in comparison with other kinds of processes, which could be mined by using process mining techniques to acquire practical SME knowledge.

4 Proposed Evaluation Framework

The proposed evaluation framework consists of two parts: 1) evaluation criteria for assessing the KM requirements (features) of SME processes (requirements related to managing SME knowledge), and 2) a knowledge flow evaluation technique.

4.1 Elicited Evaluation Criteria

Di Ciccio et al. have specified three distinguishing features for a Knowledge-Intensive Process (KIP) [Di Ciccio et al., 2015]. SME processes are knowledge-intensive since:

1. Method engineers use their knowledge, including their own experience and the knowledge provided in SME description documents, to choose the most appropriate SME process.
2. The main stages of SME process (analyzing situational requirements, choosing a SME method, applying the method), are knowledge-driven.
3. SME processes should be flexible to support the: 1) "Ability to construct various styles of methodologies", and 2) "Ability to provide a basic method that can morph in time" [Henderson-Sellers et al., 2014].

A SME method should encompass all the characteristics required for managing both general and special aspects of SME knowledge. We have analyzed these characteristics, and elicited suitable criteria to evaluate these characteristics. In total, 381 criteria were elicited over the course of two years. The criteria fall into three categories: KIP-Specific, SDM-Specific and SME-Specific. The following paragraphs discuss the characteristics of and the method used for eliciting each of these categories.

1. KIP-Specific: These criteria scrutinize the features that should be possessed by all KIPs (examples are shown in Table 1). Since both SDM and SME processes should support the use of knowledge, the criteria in this category should be satisfied by the process parts of the SME methods as well as the SDMs that they produce. In total, 106 criteria were elicited in this category. The third column in Table 1 presents the measures by which SDMs can be assessed as to their support for knowledge-intensive features, and the fourth and fifth columns provide the corresponding criteria for assessing SME methods. Thus, each row of the table shows a pair of criteria for evaluating SME methods as to a specific feature. There are two subcategories:
    a. Ability to generate a KIP: The output of an SME method is an SDM; thus, an SME process is a kind of KIP that itself produces another KIP. The criteria within this subcategory are intended to evaluate this feature, and have been elicited based on measures for assessing knowledge-intensive features in SDMs (shown in the third column of Table 1). Instances of these criteria are shown in the fourth column.
    b. Ability to support a KIP: These criteria (shown in the fifth column of Table 1) are of two types, based on the feature that they assess: 1) features that a process should possess as a KIP, and 2) features necessary to manage the knowledge-related aspects of a process.
2. SDM-Specific: These criteria are aimed at assessing the capability of SME methods to generate SDMs that themselves are capable of managing SDP knowledge (examples are shown in Table 2); thus, resources on KM in software engineering have been useful for extracting this category of criteria. In order to elicit these criteria, the knowledge-intensive features of SDMs have first been elicited, and 194 criteria have then been identified for assessing these features. The elicited criteria help assess the process/ML part of the SME method. As shown in Table 2, these criteria assess the following features of the SME process: 1) The features required to generate an SDM, which is itself capable of managing the SDP/ML knowledge (fourth column), and 2) The

features of an SME method as a special kind of SDM (fifth column); SME processes too should exhibit the features of an SDM that are required in support of the KM process.

3. SME-Specific: SME methods produce SDMs, yet since "Software processes are software too" [Osterweil, 2011], an SME method itself is considered a special kind of SDM. As such, SME methods and SDMs are of the same structure; they are both made up of two parts, a process and a modeling language. In addition, if a specific measure can be used for assessing a feature in an SDM, the same measure can be adapted for assessing the feature in SME methods. SME-Specific criteria (a total of 81 criteria) are aimed at assessing the capability to support the knowledge-intensive features of SME methods as to both process and modeling language (examples are shown in Table 3).

Due to limitations in space, the complete sets of criteria have been made available online in a supplementary file [Dehghani and Ramsin, 2023-2]. We have provided the subsets shown in Tables 1 to 3 to facilitate the discussions made in the evaluation section, where the strengths and weaknesses of SME methods will be presented. It is worth mentioning that since SME-specific and SDM-specific criteria assess the capability to embed a KM process in SME methods, the resources used for eliciting knowledge-intensive features (some of which are mentioned in Tables 2 and 3) were selected with the following conditions in mind:

1. Resources should describe knowledge-intensive features that require making knowledge-driven decisions throughout the SME process.
2. Resources should address general features, rather than features of specific paradigms and approaches. For example, knowledge-intensive features of goal-driven SDMs have not used been used for eliciting the criteria.
3. Resources should provide comprehensive and unambiguous explanations to describe the features, rather than just providing high-level outlines.

Criterion ID	**KIP Feature Being Assessed**	**Corresponding Measure for Assessing SDMs**	**Corresponding Criterion for Assessing the SME Process (SDM-Based)**	**Corresponding Criterion for Assessing the SME Process (KIP-Based)**
KIP 57,58	Clarity of the external events that affect the KIP process [França et al., 2015]	Specification of the external events that affect the SDM process	Providing mechanisms for specifying the external events that affect the SDM process	Specification of the external events that affect the SME process
KIP 75, 76	Supporting unpredictable events [Mundbrod and Reichert, 2017]	Support for unpredictable events in the SDP	Providing mechanisms to support unpredictable events in the SDP	Supporting unpredictable events within the SME process

Table 1: Examples of KIP-Specific Criteria

Criterion ID	Knowledge-Intensive Feature of the Process/ML Part of the SDM	Corresponding Measure for Assessing SDMs	Corresponding Criterion for Assessing SME Method (SDM Builder Viewpoint)	Corresponding Criterion for Assessing SME Method (SME Viewpoint)
SDM-P 9, 10	Similarity of atomic method parts to knowledge sharing components [Henderson-Sellers et al., 2014]	Using reusable method parts that support SDP knowledge	Forcing the provision of reusable method parts that support SDP knowledge	Forcing the provision of reusable method parts that support SME process knowledge
SDM-ML 1, 2	Support for preserving modeling logic [Ramsin, 2006]	Support for modeling logic	Forcing the modeling of the semantic aspects of the SDM process	Forcing the modeling of the semantic aspects of the SME process

Table 2: Examples of SDM-Specific Criteria

Criterion ID	Knowledge-Intensive Feature of the Process/ML Part of SME	Corresponding Criterion for Assessing the SME Method
SME-P 25	Capability to manage situational factors (based on [Little and Deokar, 2016])	Support for identifying new situational factors
SME-ML 3	Support for multi-dimensional modeling [García-Borgoñon et al., 2014]	Support for modeling the various aspects of knowledge: processes, products, & producers

Table 3: Examples of SME-Specific Criteria

4.2 Proposed Technique for Evaluation of Knowledge Flow in SME Methods

This technique provides a simple analysis method for comparing the “To-Be” and “As-Is” capabilities in establishing appropriate knowledge flows. The “To-Be” situation is first provided; then, the method for analyzing the “As-Is” situation is explained.

4.2.1 Envisioned “To-Be” Situation

Figure 2 shows the three types of knowledge flows desired to be established by SME methods. The flows are as follows:

- The arrows drawn between the white triangles refer to the knowledge flows that are currently established. As an example, most traditional methods have considered the effect of team size on engineering SDMs. In other words, “paying attention to team size” is a knowledge content (situational factor) that has previously been shared (pointed out) by creators of SME methods.
- The arrows drawn between the shaded triangles indicate the knowledge flows that are required to be established in future. For example, the effect of new situational factors that are found throughout the SDP should be shared.
- The diagonal arrows show the kind of knowledge flow that should be maintained. For example, in some contexts, team size is a situational factor that is related to interchangeability of responsibilities. This fact should be shared as a new update to the previously shared knowledge.

4.2.2 Analysis of “As-Is” Situation

A two-step process is proposed to extract and illustrate information about the "As-Is" state of knowledge flows. These steps are as follows:

1. Extraction: This step is aimed at extracting information about the path of movement for each of the knowledge contents that are produced (or used) throughout the SME process. The term “information” encompasses data about

time, place, and content aspects of knowledge flow, and "knowledge content" may refer to people (producers), or products, or processes. The following paragraphs provide detailed information about these concepts and explain the method for information extraction. Based on the main goal of KM processes, that is: "distribution of the right knowledge from the right people to the right people at the right time" [Riege, 2005], the following components have been identified for establishing knowledge flows in SME process:

a. Knowledge Content: Refers to the knowledge that is transferred throughout the SME method. This concept is called "Method Knowledge" and can be categorized in three main classes: "Process", "Product", and "Producer" [Henderson-Sellers et al., 2014].
b. Transfer Time: Indicates the time at which knowledge is transferred.
c. Transfer Place: Specifies the destination to which knowledge should be flowed, and also the source from which knowledge is extracted and transferred. A transfer place (source/destination) can be of three types: Individual, Location, and Artifact. For instance, a method engineer is an individual who shares his/her experience, a lab is a location where individuals share their knowledge, and post-mortem documents are artifacts for sharing the lessons learned from a project.

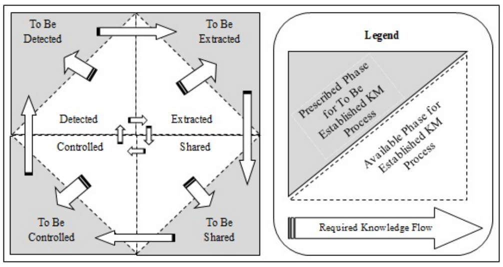

Figure 2: Essential Knowledge Flows in SME Methods

It should be noted that Product and Artifact are different concepts: An Artifact is a facility prescribed for sharing knowledge (like a groupware or a document), whereas a Product is the knowledge content that is produced in the SME process. To extract the above-explained information, a table should be produced for each knowledge content (product, process, and producer). This table provides information about the time and place in which a specific type of knowledge content is transferred. The column headings refer to the place to which (or from which) the knowledge is transferred, and the row headings specify the time at which knowledge is shared. Each cell is filled with the knowledge content that is transferred at the time and place that is specified by the corresponding row and column. As an example, the main time intervals for transferring knowledge in the Assembly-Based method are its main process steps, and the place to which (and from which) knowledge is transferred is an individual (the method engineer) or an artifact (repository of method chunks or list of project characteristics). Table 4 and Table 5 provide information about the product, process, and producer aspects of knowledge flow in this method. For example, as seen in Table 4, at the end of the "Select Method Chunks" activity, method chunks are transferred to the repository.

2. Integration: The information gathered in the previous step should be integrated and illustrated. To this aim, the tables produced in the previous step are analyzed and the new information is shown in a new table. This table contains information about the turnover of all knowledge contents. Then, by using the information provided in this table, the capability to support each of the arrows shown in Figure 2 (essential knowledge flows) should be investigated. Figure 3 and Table 6 show examples for the Assembly-Based method.

Time \ Place		Individual	Location	Artifact	
		Method Engineer	✖	Repository of Method Chunks	List of Project Characteristics (in Four Dimensions)
End of Main Steps	*Specify Method Requirements*	Requirements List	NULL	NULL	NULL
	Select Chunks	NULL	NULL	Method Chunks	NULL
	Assemble Chunks	Method Parts	NULL	NULL	NULL
	Specify Project Characteristics	Project Characteristics	NULL	NULL	Project Characteristics
	Stop SME Process	NULL	NULL	NULL	NULL

Table 4: Product Aspect of Knowledge Flow in Assembly-Based

Time \ Place		Individual	Location	Artifact	
		Method Engineer	✖	Repository	List of Project Characteristics
End of Main Steps	*Specify Method Requirements*	*Process:* Intention-driven requirements specs, Process-driven requirements specs; *Producer*: NULL	NULL	NULL	NULL
	Select Chunks	*Process:* Decomposing, aggregating, refining, verifying, and evaluating chunks, Provision of PC-driven process map, Requirements-driven chunk selection; *Producer*: NULL	NULL	NULL	NULL
	Assemble Chunks	*Process*: Integrating chunks, Associating chunks; *Producer*: NULL	NULL	NULL	NULL
	Specify Project Characteristics	*Process:* Project characterization, Refining method chunks; *Producer*: NULL	NULL	NULL	*Process:* Project characterization *Producer:* NULL
	Stop SME Process	*Process:* Validating method completeness; *Producer*: NULL	NULL	NULL	NULL

Table 5: Process & Producer Aspects of Knowledge Flow in Assembly-Based

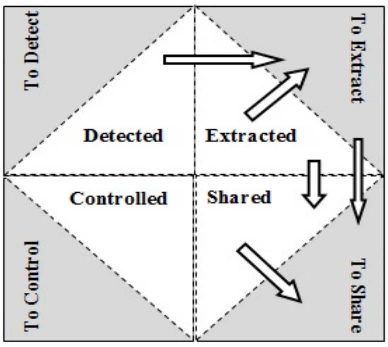

Figure 3: KM Process Schema (original state) in the Assembly-Based Method

KM Process Phase \ Knowledge Content		Product/Artifact	Process & Producer
Detection	*Detected*	Repository of Method Chunks	NULL
	To Detect	NULL	NULL
Extraction	*Extracted*	General (Dimensions of) Project Characteristics	*Process:* Intention-driven requirements specification; Process-driven requirements specification; Decomposing, aggregating, refining, verifying, and evaluating method chunks; Creating PC-Driven process map; Requirements-driven selection of method chunks; Integrating method chunks; Associating method chunks; Project characterization *Producer:* NULL
	To Extract	Project Characteristics, Requirements List, Method Chunks, Method Parts	*Process:* Validating method completeness, Project characterization; *Producer:* NULL
Sharing	*Shared*	All "Extracted" products	*Process:* Extracted processes; *Producer:* NULL
	To Share	All "To Extract" products	NULL
Controlling	*Controlled*	NULL	NULL
	To Control	NULL	NULL

Table 6: Knowledge Content-Based Analysis of the KM Process: Assembly-Based

5 Proposed KM-Driven Improvement Model

We have proposed an improvement model that can be used for alleviating the weaknesses that stem from the failure to manage SME process knowledge. Before describing the model itself, the constituents of the proposed model are explained.

5.1 Model Constituents

The constituents of our proposed model for improving SME processes have been inspired by the research conducted in [Team–CMMI Product, 2010], which has proposed a Capability Maturity Model Integration (CMMI) model for improving product/service development processes. Our proposed model provides a set of integrated guidelines for employing appropriate KM practices so as to alleviate KM-related weaknesses in SME methods.

Our proposed model is composed of three main constituents (as seen in Figure 4):

1. Problem Area: Specifies the problem for which a solution should be found. A problem can be seen as a requirement (or goal) that has not been satisfied. Thus, it corresponds to a subset of the requirements (criteria) of the proposed evaluation framework. The following elements describe the problems:
    a. Problem Sub-Area: Determines the sub-area to which the problem is related, and is one of these general SME process steps: "Approach of Method Construction", "Validation of Completeness", "Evaluation", and "Construction Strategy" [Henderson-Sellers et al., 2014].
    b. Goal: Describes the problem by specifying the goal for resolving the problem and also the goal for improving the current situation.
3. Solution Area: Provides the suggested solution, which is itself composed of a number of practices categorized according to the four phases of the KM process (Detection, Extraction, Sharing, and Controlling); in other words, problems are solved through detecting the appropriate knowledge resources, extracting and sharing the required knowledge contents, and/or controlling the quality of the KM process. Since the KM practices used in SME are inadequate, we have conducted an interdisciplinary research [Repko and Szostak, 2020] to reuse KM practices from other domains. For example, we propose the reuse of certain tried and tested KM practices used in Product Lifecycle Management (PLM) [Ameri and Dutta, 2005, Stark, 2020].
2. Explanation Area: Provides the detail about the constituents of the problem and solution areas through the following elements:
    a. Related area: Specifies the requirements/solutions that relate to the intended problem/solution.
    b. Description: Describes the problem/solution.
    c. Example: Provides an example to clarify the problem/solution.
    d. Conditions: Specifies the conditions that should be satisfied for using the proposed solution to achieve promotion and resolution of goals.

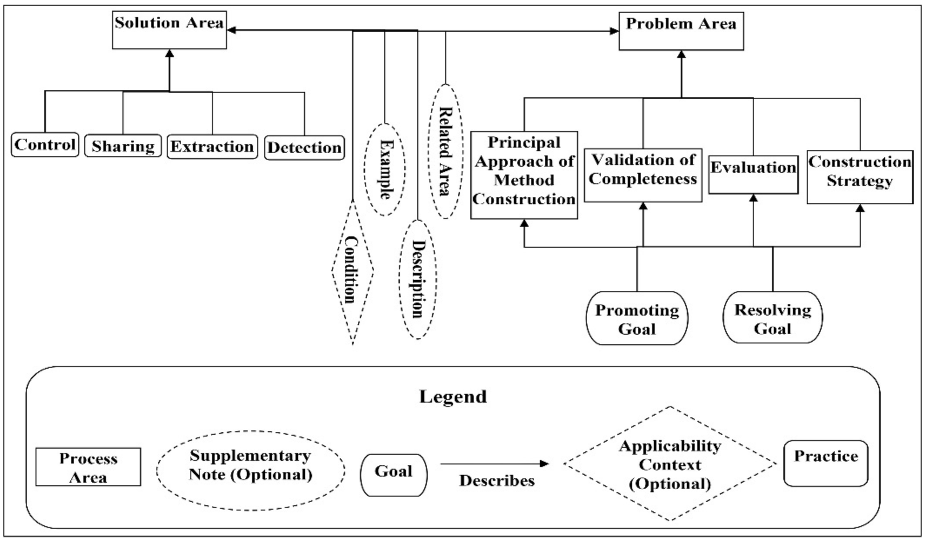

Figure 4: Constituents of the Proposed Improvement Model (inspired by [Team–CMMI Product, 2010, Marjanovic and Freeze, 2012])

5.2 Description of Proposed Improvement Model

The proposed model provides specific solutions in defined circumstances for satisfying the requirements in the proposed evaluation framework (as explained in the previous section). This subsection describes the model by providing an example of the solution proposed to improve the Hybrid method proposed by [Ramsin, 2006].

As shown in Figure 5, the example refers to a problem identified by the KIP-6 and KIP-5 evaluation criteria (as indexed in the evaluation framework), which refer to the inability in explicit provision of knowledge within SME and SDM processes. Modelling the process knowledge helps alleviate these weaknesses by preserving and sharing method engineers' and software developers' knowledge. Therefore, the EPC notation [Thomas and Fellmann, 2009] has been proposed for modelling processes. As seen in Figure 5, familiarity with EPC notation has been considered as a prerequisite for applying the proposed solution. Since the Hybrid method has not considered the effect of various fine-grained events on choosing SME strategies, this notation has been chosen to help share knowledge about events and corresponding process flows.

Problem: KIP 6
Description: A method is required for modeling the SME process knowledge.
Related Area: KIP 5
Example: Hybrid Methodology (Ramsin 2006) suffers from this problem.

Problem Sub-Area: This problem is in relation to all four introduced sub-areas because modeling helps achieve a deeper understanding of the situation at hand. This affects selecting the construction strategy and approach, validating the completeness, and evaluating the engineered method.

Resolving Goal: Modeling all dimensions of SME process knowledge (KIP 6)
Promoting Goal: Modeling all dimensions of software development process knowledge (KIP 5)

Solution:

Solution Sub-Area: Knowledge Sharing

Condition: Familiarity with EPC (Event-Driven Process Chain) notation (Thomas and Fellmann MA 2009)
Description: Use the EPC notation for modeling SME process
Related Area: -
Example:

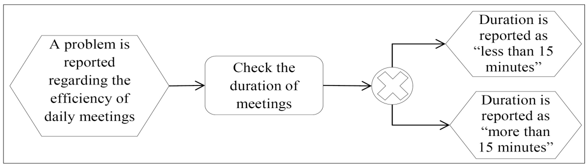

Figure 5: Example of Proposed Solutions

6 Evaluation of the Proposed Framework and Model

We have theoretically and empirically evaluated the proposed evaluation framework and improvement model, the results of which are discussed in the following subsections. An analysis of the threats to the validity of the evaluation results is provided in the final subsection.

6.1 Theoretical Evaluation

We applied the proposed framework and model to eight SME methods. The following subsection introduces these methods. Thereafter, the evaluation results are provided for the proposed framework and model.

6.1.1 Review of SME Methods

Eight SME methods have been selected for evaluation [Ramsin, 2006, Mirbel and Ralyté, 2006, Aharoni and Reinhartz-Berger, 2008, Serour and Henderson-Sellers, 2004, Seidita et al., 2007, Bajec et al., 2007, Hoppenbrouwers et al., 2008, Henderson-Sellers et al., 2014]. These methods have been selected based on the availability of description documents, and adequacy of the available documentation (as to richness in detail) for conducting proper evaluation. Furthermore, we have used the framework provided in [Nehan and Deneckere, 2007] for comparing and selecting superior methods based on the following features: nature, construction technique, knowledge representation technique, supported dimension (process/product), abstraction approach, support for formalism, flexibility, knowledge construction approach, and knowledge structuring method. For sake of brevity, only Henderson-Sellers et al.'s assembly-based method will be further explained herein.

Henderson-Sellers et al. have listed the following seven approaches for constructing an SDM: Assembly-Based, Extension-Based, Deontic Metrics, Activity Diagrams, Ad Hoc, Configuration-Based, and Paradigm-Based [Henderson-Sellers et al., 2014]. Based on the evaluation results provided in [Seidita et al., 2007], and also through comparing these strategies, we have realized that the assembly-based approach is more powerful than other approaches in establishing the required knowledge flows.

Therefore, we have selected the extended version of the assembly-based approach for discussion. This version provides a requirements-based process for selecting and assembling the appropriate method chunks [Kornyshova et al., 2007]. The requirements are inspired by project characteristics which are categorized into four dimensions, namely: Organizational, Human, Application Domain, and Development Strategy [Henderson-Sellers et al., 2014]. As shown in Figure 6, this method prescribes a five-step process:

1. Specifying Project Characteristics: Investigate the project characteristics within the four categories provided.
4. Specifying SDM Requirements: Specify the requirements through asking development and management team members to participate in requirements elicitation activities.
5. Selecting Method Chunks: Select the appropriate chunks from the method chunk repository based on their weights and priorities.
6. Assembling Method Chunks: Assemble the chunks into the target SDM.
7. Stopping SME process: Stop the process if requirements are satisfied.

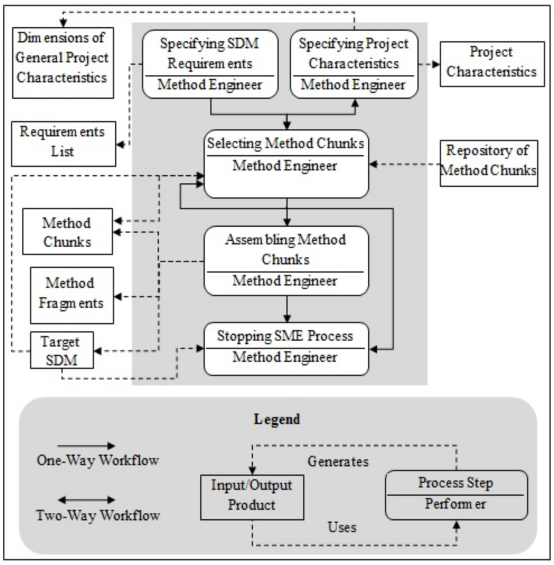

Figure 6: Assembly-Based Method (original state, derived from [Dehghani and Ramsin, 2023-1])

6.1.2 Theoretical Evaluation of Proposed Framework

The evaluation framework was first evaluated by applying a set of meta-criteria. Then, its two constituents were assessed. The results are thus respectively explained within three categories of capabilities, as follows.

1. **Capability to satisfy meta-criteria:** The framework has been iteratively assessed and improved based on the following meta-criteria, derived from [Taromirad and Ramsin, 2008, Dehghani and Ramsin, 2015]:
    a. Coverage of the main constituents of KM and SME processes: The proposed knowledge flow evaluation technique helps assess the transfer of the right knowledge content to the right method engineer at the right time (constituents of the KM process). Furthermore, the criteria assess the flow of all types of method engineering knowledge (as to process, people, and products [Ramsin and Paige, 2008]) that are embedded in the SME process.
    b. Precision: Production of detailed, clear, and explicit evaluation results has been ensured by: 1) specifying the bases for eliciting the criteria and defining the different levels of values that may be assigned to the evaluation criteria, and 2) specifying the three types of knowledge flows that should be assessed.
    c. Simplicity: The framework is easily understood because: 1) the criteria are traceable to tangible features and the value ranges are precisely defined, and 2) the proposed knowledge flow evaluation method uses a simple table-based method for analysis.
    d. Consistency: In order to provide consistent criteria, complementary sources have been used for eliciting the criteria, and all

inconsistencies have been identified and resolved. Besides, the proposed knowledge flow evaluation technique is consistent with the main constituents of SME process (people, product, and process).

e. Minimum overlapping: Overlaps have been identified and minimized by considering features that are common to SME and KM processes.
f. Generality: The framework can be used to evaluate all SME methods.
g. Balance: As the framework has been proposed through analyzing both the applications and the definitions of SME methods, descriptive and practical aspects are both covered.
h. Conformance with KM foundations: The criteria are consistent with KM foundations because we have considered the knowledge-intensive features of the SME process and SDP.

2. **Capability to reveal strengths and weaknesses in support of KM-Driven SME requirements:** In order to determine the levels to which existing SME methods satisfy the elicited KM requirements, a criteria-based evaluation has been conducted. For this purpose, three levels of satisfaction have been specified: 1) Satisfaction (Full), 2) Partial Satisfaction, and 3) Non-satisfaction. The complete evaluation results are provided in [Dehghani and Ramsin, 2023-2]. As an example of the detailed results, a small subset of the evaluation results for the Assembly-Based method is provided in Table 7; furthermore, for some criteria, a subset of the weaknesses and strengths revealed for this method is provided in Table 8. To summarize the results, the following items have been revealed as some of the strengths/weaknesses common to all the methods:

   Strengths

   a. Flexibility in changing knowledge flow features (time/place/content)
   b. Simplicity in using the models and processes prescribed
   c. Provision of basic knowledge contents

   Weaknesses

   a. Inability to share process dependency knowledge
   b. Neglecting the knowledge required for monitoring the SME process
   c. Inability to manage communications of method engineers/customers
   d. Ignoring the knowledge required for continuous engineering of method requirements
   e. Inability to manage the cultural issues that facilitate knowledge sharing in the SME process
   f. Lack of support for providing the appropriate knowledge for preventing misunderstandings
   g. Lack of attention to discovering/establishing new knowledge flows

3. **Capability to reveal strengths/weaknesses in knowledge flows:** Results of evaluating knowledge flows in the assembly-based approach were provided in Section 4.2. The final results of knowledge flow evaluation in the remaining seven SME methods are shown in Table 9. The detailed evaluation results are provided in [Dehghani and Ramsin, 2023-2]. Most of the methods neglect the need for continuous update of existing knowledge flows. For example, Ramsin and Paige have provided a list of general requirements that should be satisfied by all the SDMs produced, but they have neglected the need to update this list [Ramsin, 2006]. Also, establishing new knowledge flows has been neglected;

for instance, Mirbel and Ralyté have introduced two knowledge producers (method engineer and ISD crew member), but no mechanisms have been prescribed for extracting/sharing their knowledge [Mirbel and Ralyté, 2006].

Criterion ID	Evaluation Result	Reason
KIP 2	✓ (Satisfaction)	The process steps are rigid, but the method engineer can decide about certain factors such as requirements, project characteristics, and the assembly method.
SDM-P 2	✖ (Non-satisfaction)	Updating the SME process is not forced.
SDM-ML 4	- (Partial Satisfaction)	Interactions of method chunks are managed, but managing the interactions of finer-grained elements (such as products) is not forced.

Table 7: Examples of Evaluation Results: Assembly-Based Method

KIP- Specific Criteria	SME-Specific Criteria	SDM-Specific Criteria
Support for change management by providing two paths (requirements-based and project-characteristics-based) for selecting method chunks (KIP 28)	Managing the knowledge required for reusing method chunks (SME-ML 21)	Support for extracting knowledge through sharing new method chunks (SDM-P 88)
Lack of support for managing the knowledge hidden in communications with method customers (KIP 91)	Inability to provide knowledge required for producing testable products (SME-ML 29)	Need for appropriate techniques for extracting method engineers' knowledge (SDM-P 172)

Table 8: Examples of Revealed Strengths and Weaknesses

Method, Content (KM Process)		Detecting		Extracting		Sharing		Controlling	
		Detected	*To Detect*	*Extracted*	*To Extract*	*Shared*	*To Share*	*Controlled*	*To Control*
Ramsin and Paige	❶	□	□	■	■	■	■	□	□
	❷	□	□	■	□	■	□	□	□
	❸	□	□	□	□	□	□	□	□
Mirbel and Ralyté	❶	⊡	□	■	◘	⊡	■	□	□
	❷	□	□	■	◘	■	■	□	□
	❸	⊡	□	□	□	□	□	□	□
Aharoni and Reinhartz-Berger	❶	□	□	■	⊡	■	□	□	□
	❷	□	□	□	□	□	□	□	□
	❸	□	□	□	□	□	□	□	□
Henderson-Sellers and Serour	❶	□	□	■	□	■	□	□	□
	❷	□	□	□	□	□	□	□	□
	❸	■	□	□	□	□	□	■	■
Seidita et al.	❶	□	□	□	⊡	□	□	□	□
	❷	□	□	□	⊡	□	□	□	□
	❸	□	□	□	□	□	□	□	□
Bajec et al.	❶	□	□	■	■	■	■	□	□
	❷	□	□	□	⊡	□	□	□	□
	❸	□	□	□	□	□	□	□	□
Hoppenbrouwers et al.	❶	□	□	□	⊡	□	□	□	□
	❷	□	□	□	⊡	□	□	□	□
	❸	⊡	⊡	□	⊡	□	□	□	□

Legend:
❶: Product/Artifact Knowledge Content; ❷: Process Knowledge Content; ❸: Producer Knowledge Content
■: All the prescribed knowledge contents are (will be) transferred to support the KM process.
□: No knowledge content is (will be) prescribed in this phase.
◘: Some (not all) of the prescribed knowledge contents are (will be) transferred to support the KM process.
⊡: None of the prescribed knowledge contents are (will be) transferred to support the KM process.

Table 9: Results of Knowledge Flow Evaluation in Seven SME Methods

6.1.3 Theoretical Evaluation of Proposed Improvement Model

We applied the proposed improvement model to the SME methods evaluated. This assessment has shown that the model is capable of establishing all the knowledge flows required. Since it is not possible to present the complete results herein, an overview of the improved version of the Assembly-Based method will be provided as an instance. The original state of this method was previously explained (Figure 6). As shown in Figure 7, the improved iterative-incremental process consists of four main phases (shown in the shaded rectangles): 1) The first phase satisfies the preconditions required for managing the process knowledge provided, and helps decide whether to improve an available method or engineer a new one; 2) The new parts of the target SDM are then constructed through performing the stages that are shown in the left-side rectangle; and 3) The improved parts are produced through performing the right-side phase; 4) Finally, it is verified that the engineered method satisfies all the requirements and, if required, a new iteration is performed to construct new method parts and also to maintain the quality of the parts constructed. The activities that are shown on the arrow are umbrella activities that should be performed throughout the whole process to satisfy managerial criteria. The KM process, shown at the center of the figure, is performed in parallel with all other phases to ensure proper elicitation/maintenance of SME process knowledge. In order to depict the knowledge flows thus established, Figure 8 and Table 10 provide examples of the solutions used for establishing each of the required knowledge flows in the assembly-based approach. Also, for each of the improved SME methods, an example of the alleviated weaknesses has been presented in Table 11.

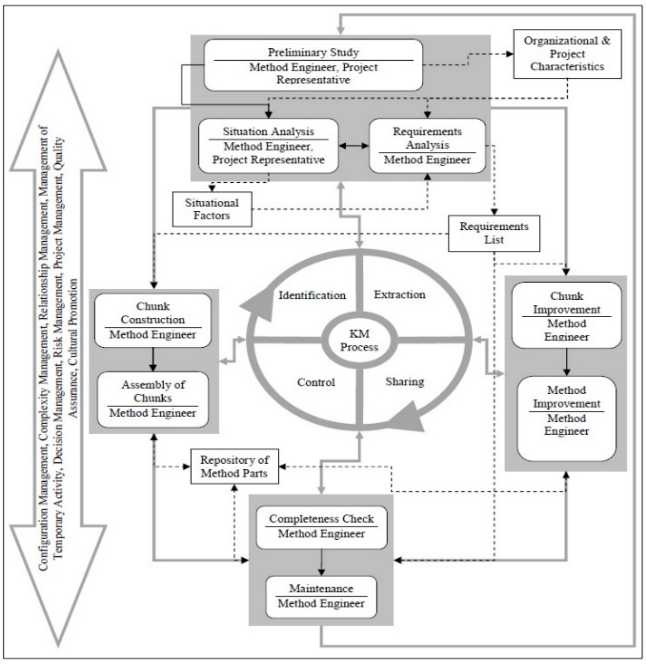

Figure 7: Assembly-based Method (improved state)

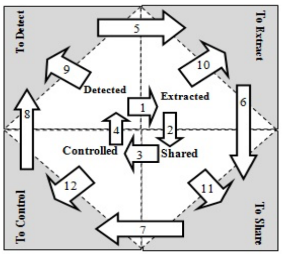

Figure 8: KM Process Schema (improved state) in the Assembly-Based Method

Knowledge Flow ID (in Figure 8)	Problem ID	Problem	Condition	Solution	Relevant Constituent
1	SDM-P 136	Need for identifying and documenting method engineers' knowledge	SME process is people-intensive	Mine human resources data to distinguish among experience, expertise, and specialty [Palshikar et al., 2011]	New Technique
2	KIP 50	Support for transferring detected knowledge content	Knowledge contents are in Internet resources	Use tagging mechanisms [Karni and Levy, 2014]	New Technique
3	SME-ML 11	Need for modeling control flow knowledge	Need to control knowledge sharing	Build knowledge flow views [Liu and Lin, 2012]	Modeling process steps
4	KIP 16	Need for tracing and documenting event-driven decisions	Contextual understanding is important	Deconstruct, integrate, combine, connect, and internalize knowledge blocks so as to provide double-loop learning [Evans et al., 2015]	Final process step
5	SDM-P 81	Forcing the method engineers to update in-use technologies throughout the SDP	Need for continuous learning	Force the creation of a learning environment [Easterby-Smith and Lyles, 2011]	All process steps
6	KIP 108	Providing cultural infrastructure for continuous knowledge sharing	SME process is performed by a small group	Assess the effect of various factors at various time intervals [Lin, 2014]	First process step
7	KIP 100	Need for preventing unauthorized process copying	Activities support competitive advantage	Manage the knowledge hidden in activities [Meihami and Meihami, 2014]	All process steps
8	SDM-P 11	Need for managing upcoming risks	Need for clarifying scope	Use combination techniques [Neves et al., 2014]	All process steps
9	SME-ML 16	Need for domain knowledge	Need to understand SDM requirements	Use ontology-based modeling techniques [Kaiya and Saeki, 2006]	First process step
10	SDM-P 52	Need for updating SME method requirements	Continuous change of requirements	Manage the requirements as a kind of knowledge [Botha et al., 2014]	First process step
11	SDM-P 6	Need for sharing knowledge in the SME process	The knowledge should be shared informally	Use interactive information technologies [Davison et al., 2013]	All process steps
12	SDM-P 46	Need for managing strategies	KM strategy is not clear	Choose suitable KM strategy [Kim et al., 2014]	First process step

Table 10: Example of Applying the Proposed Improvement Model to Assembly-Based

Method	Weakness	Condition	Solution	Corresponding Constituent
Henderson-Sellers et al.'s Assembly-Based Approach	Need for appropriate techniques to extract method engineers' knowledge (SDM-P 172)	Experts communicate via online facilities	Use the algorithm provided in [Wang et al., 2013]	New Technique
Ramsin and Paige	Inability to force the embedding of knowledge sharing mechanisms in the produced SDM (SDM_P 59)	Lack of attention to prerequisites for knowledge sharing	Study knowledge sharing prerequisites such as trust [Park and Lee, 2014] and culture [Lee et al., 2016]	A phase is added to establish knowledge sharing prerequisites
Mirbel and Ralyté	Lack of support for modeling the various dimensions of SME process knowledge (SME-ML3)	Need for human knowledge	Use human-centered knowledge modeling techniques [Leake et al., 2014]	New Technique
Aharoni and Reinhartz-Berger	Inability to consider the policies that affect the SME process (SDM-P 30)	Method is to be used in public-sector organizations	Involve SME stakeholders in specifying and updating organizational policies [Riege and Lindsay, 2006]	New Guideline
Serour and Henderson-Sellers	Lack of support for preventing the SME process from being dependent on environmental factors (SME-P 27)	The method environment is dynamic	Plan for maturing the dynamic capabilities of the organization [Li and Liu, 2014]	A planning phase is added as the first phase of the SME process
Seidita et al.	Inability to model control flow knowledge (SME-ML 11)	Knowledge flow is changeable	Use the petri-net-based model proposed in [Wang and Wang, 2016]	A modeling task is added, in parallel to all the phases
Bajec et al.	Lack of support for encouraging method engineers to share their knowledge (SDM-P 76)	Teamwork is not recognized as a value	Use the team management techniques provided in [Katzenbach and Smith, 2015]	Sets of techniques are added that should be used in umbrella activities
Hoppenbrouwers et al.	Inability to study cultural feasibility to manage SME process knowledge (SDM-P 98)	SME process will used in a project-based organization	Use Cameron-Quinn's framework to evaluate cultural features [Wiewiora et al., 2013]	A new phase is added as the first step of the SME process

Table 11: Examples of Alleviated Weaknesses (by applying the proposed model)

6.2 Empirical Evaluation

As mentioned, we have conducted four case studies. Four Iranian IT companies were chosen as case-study venues. In all of these companies, method engineers had previously tried to customize the Scrum methodology in accordance with the special features of their project situations, yet they still felt the need for improving their in-use method engineering and software development processes to alleviate the remaining problems. The identities of these companies will not be disclosed; however, their general features are shown in Table 12. It should be noted that we have considered three main criteria in selecting the companies: 1) Availability of the resources required for conducting the case study, 2) Need for KM-driven solutions for solving the problems encountered, and 3) "Maximum Variation" [Runeson et al., 2012] in the problems encountered.

Name	Age	Size (Respondents)	Example of Problems
A (One Unit)	25	11	Need for managing impulsive developers.
B (Four Units)	17	20	Need for managing mistrust among managers and developers.
C (One Unit)	7	15	Need for training appropriate teamwork techniques.
D (Start-Up) (One Unit)	3	3	Need for preserving the experience acquired through communicating with the customers.

Table 12: Features of Selected Companies

To collect the evaluation data, two rounds of semi-structured interviews and questionnaires were designed (provided in [Dehghani and Ramsin, 2023-2]). In the first round, we searched for the problems, and assessed the proposed framework's capability to reveal strengths/weaknesses of in-use processes. In the second round, conducted after applying the model, we investigated the opinions of developers and method engineers as to the efficiency of the solutions prescribed by the model and also the usefulness of the evaluation results. It should be noted that because of limitations in communicating with respondents and also due to the large size of the proposed evaluation framework and improvement model, questions were specifically aimed at finding the main problems and their solutions. Considering the limitations in space, some examples of the questions used in interviews and questionnaires, are provided in Table 13.

Examples for First Round of Questions (Questionnaire)					
Question (Target Respondent(s))	**Strongly Agree**	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Suggestions to use new technologies are rejected without being reasoned about (D).					
In comparison to works conducted individually, team works are less efficient as to time and quality (B).					
People are moved around to learn new skills (B).					
Specific KPIs are used to monitor the processes (M).					
Examples for Second Round of Questions (Questionnaire)					
Learning is now faster (B).					
Learning is now easier (D).					
Suggestions are now more welcome and are assessed (D).					
Reasons for reworking are now discussed periodically (M).					
Plans are now more applicable (B).					
Examples for First Round of Questions (Semi-Structured Interview)					
What are the process steps, inputs, outputs, and products throughout the software development process (B)?					
How are the processes customized for different projects (M)?					
Which responsibilities are defined (B)?					
How do you acquire new skills (D)?					
Examples for Second Round of Questions (Semi-Structured Interview)					
Do you confirm the weaknesses found (B)?					
In your opinion, are the solutions applicable (B)?					
What are the weaknesses of this case study in terms of applicability, time, etc. (B)?					
How long did it take to use the improvement model (M)?					
Legend: M: Method Engineers; **D:** Developers; **B:** Both Developers and Method Engineers					

Table 13: Examples of Questions (used in interviews and questionnaires)

6.2.1 Empirical Evaluation of Proposed Framework

Table 14 shows the results of analyzing the responses received on evaluating the proposed framework. The main strengths/weaknesses specified in this table are those features of processes that have resulted in satisfaction/nonsatisfaction of a large number of criteria. By applying the framework, we were able to find the weaknesses/strengths in support of knowledge-intensive features of in-use processes. Since the framework assesses the capability to both manage SME knowledge and engineer SDMs that support software development KM, these strengths/weaknesses have been revealed by analyzing SME methods and the SDMs that they have produced.

We also asked method engineers (project managers) to evaluate our framework by analyzing the results of applying our framework; the opinions were as follows:

1. Validity: Except for one expert, others confirmed that the framework discovered the strengths/weaknesses correctly. Although the results were positive, three issues threaten their validity: 1) the framework was only used in small-sized units, 2) due to the large number of the criteria, experts could not review them all, and 3) the framework was only used in Iranian companies, whereas diverse cultural contexts would be preferable.
2. Comprehensiveness: As mentioned before, because of the large number of criteria, we could not apply the framework comprehensively, and therefore could not demonstrate its comprehensiveness. We also realized that the results could be more comprehensive if the history behind the formation of the in-use processes was assessed; we therefore conducted this historical assessment.
3. Accuracy: Experts stated that the results were precise, as the questionnaires were focused on specific issues. All experts confirmed that not only do the results cover various knowledge aspects of a problem/opportunity, but they also help find KM-driven roots for the discovered problems/opportunities.
4. Practicality: The experts confirmed that the framework was indeed applicable to their in-use processes, but they also remarked that applying the whole framework was time-consuming. To address this issue, we have proposed a risk-based categorization of the criteria by specifying the knowledge risks that would be identified by using the criteria (Table 15). This enables the method engineers to choose those criteria that help find the riskiest knowledge threats.

Company	Strength	Weakness
A	Developers' tendency to improve in-use processes	Inability to create a friendly workplace
	Vast knowledge about knowledge resources	Inefficiency of motivational mechanisms
B	Ability to create a friendly workplace (in one unit)	Inefficiency of training mechanisms
	Speciation of measures for assessing products	Dissatisfaction with career progression
C	Avoidance of structural limitations which prevent knowledge sharing	Being dependent on specific developers
	Developers' tendency to participate in workshops	Suffering from knowledge hoarding culture
D	Being familiar with appropriate mechanisms for sharing knowledge with customers	Inability to specify criteria for evaluating processes
	Ability to create a friendly workplace	Inability to update plans and make accurate estimations

Table 14: Main Strengths and Weaknesses of In-Use Processes in Companies (derived from [Dehghani and Ramsin, 2023-1])

Category of Knowledge Risk [Durst and Zieba, 2019]	**Sub-Category of Knowledge Risk (ID)** [Durst and Zieba, 2019]	**Corresponding Criterion (to identify the risk)**
"Human"	Knowledge hiding (KR-H1)	KIP 35, KIP 36, KIP 91, KIP 92, KIP 101, KIP 102
	Knowledge hoarding (KR-H2)	KIP 6, KIP 65, KIP 66,
	Forgetting (KR-H3)	KIP 9, KIP 10, KIP 21, KIP 22, KIP 73, KIP 74,
	Unlearning (KR-H4)	KIP 45, KIP 46, KIP 105, KIP 106,
	Missing/inadequate competencies of organization members (KR-H5)	KIP 63, KIP 64, KIP 79, KIP 80,
"Technological"	Risk of hacker attacks (KR-T1)	
	Risks of old technologies (KR-T2)	KIP 27, KIP 28, KIP 71, KIP 72,
	Risks of social media (KR-T3)	
	Digitalization risks e.g. IoT (KR-T4)	
"Operational"	Continuity risk (KR-Op1)	KIP 1, KIP 2, KIP 3, KIP 4, KIP 99, KIP 100,
	Knowledge gaps (KR-Op2)	KIP 29, KIP 30
	Espionage (KR-Op3)	KIP 67, KIP 68,
	Knowledge waste (KR-Op4)	KIP 13, KIP 14, KIP 65, KIP 66,
	Merger&Acquisition risk (KR-Op5)	KIP 19, KIP 20, KIP 25, KIP 26, KIP 43, KIP 44, KIP 85, KIP 86,
	Risk of using unreliable information (KR-Op6)	KIP 15, KIP 16, KIP 39, KIP 40, KIP 41, KIP 42,
	Obsolete knowledge risk (KR-Op7)	KIP 17, KIP 18
	Risk of improperly applying knowledge (KR-Op8)	KIP 7, KIP 8, KIP 33, KIP 34, KIP 87, KIP 88, KIP 97, KIP 98, KIP 103, KIP 104,
	Integration risk (KR-Op9)	KIP 23, KIP 24, KIP 53, KIP 54, KIP 83, KIP 84,
	Outsourcing risks (KR-Op10)	KIP 55, KIP 56, KIP 57, KIP 58,
	Knowledge transfer (KR-Op11)	KIP 37, KIP 38, KIP 47, KIP 48, KIP 49, KIP 50, KIP 81, KIP 82, KIP 109, KIP 110,
	Relational risk (KR-Op12)	KIP 59, KIP 60, KIP 93, KIP 94, KIP 111, KIP 112
	Knowledge acquisition risk (KR-Op13)	KIP 31, KIP 32, KIP 61, KIP 62, KIP 75, KIP 76, KIP 89, KIP 90, KIP 95, KIP 96,
	Communication risk (KR-Op14)	KIP 11, KIP 12, KIP 51, KIP 52, KIP 69, KIP 70, KIP 77, KIP 78, KIP 107, KIP 108

Table 15: Correspondence between Knowledge Risks and Evaluation Criteria

6.2.2 Empirical Evaluation of Proposed Model

We aimed to find the answer to the following two questions from the practitioners' point of view: 1) What are the strengths and weaknesses of the proposed model? and 2) Can it be claimed that the proposed model helps improve knowledge management in SME methods and SDMs? The results of analyzing the responses were as follows:

1. First Question: Most method engineers expressed that the most important strength of the proposed model was its novel approach wherein problem-driven solutions are prescribed to embed KM techniques in SME processes. However, they reported that the model failed to specify an order for applying the techniques prescribed. This weakness was alleviated after prioritizing the techniques by identifying their corresponding knowledge risks (Table 15).
2. Second Question: In order to assess the efficiency of the solutions prescribed, we first defined Likert-type scales [Brown, 2011] to quantify the respondents' answers (Strongly Agree (2), Agree (1), Neutral (0), Disagree (-1), Strongly Disagree (-2)); the data was then analyzed to calculate the following measures:

a. Precision: Our proposed solutions were prescribed based on the context in which they should be applied [Thompson and Walsham, 2004]. Staff with SME responsibilities have confirmed the efficiency of the proposed solutions in their companies. The developers' viewpoints were also analyzed as they were charged with applying the improvements. The experts' (method engineers') responses were considered as the right (true) solutions, and we then computed the ratio of the true solutions confirmed by the developers (by Formula 1). As shown in Table 16, the efficiency of most solutions was confirmed (the precision is greater than 0.5 in all the companies).
b. Mean value: Table 16 also shows the mean values of responses (calculated by Formula 2). As seen in this table, the solutions are averagely confirmed by both developers and experts.

Formula 1. Precision of Proposed Solutions

$$\frac{\sum_{\mathrm{Si}=1}^{\mathrm{Si}=Number\ of\ Solutions} C(\mathrm{Si})}{Number\ of\ Solutions}$$

$$, C(\mathrm{Si}) = \begin{cases} 1, & Consistency\ between\ developers'\ and\ experts'\ responses \\ 0, & Inconsistency\ between\ developers'\ and\ experts'\ responses \end{cases}$$

Formula 2. Mean Value of Responses

$$\frac{\sum_{\mathrm{Qi}=1}^{\mathrm{Qi}=Number\ of\ Questions} \frac{\sum_{\mathrm{Ri}=1}^{\mathrm{Ri}=\mathrm{Number}\ of\ Respondents} Response\ in\ Likert's\ Scale\ (\mathrm{Qi})}{2 * Number\ of\ Respondents}}{Number\ of\ Questions}$$

Measure \ Company		A	B	C	D
Mean	**Developers' Questions**	0.6	0.6	0.57	0.79
	Experts' General Questions	0.75	0.5	0.87	0.58
	Experts' Special Questions	0.7	0.36	0.68	0.43
Precision		1	0.67	1	0.83

Table 16: Results of Evaluating the Proposed Model

6.3 Threats to Validity

The following issues have been identified as threats to the validity of this work:

- Construct Validity: The interviews have not covered all of the criteria, and misinterpretations might have occurred in analyzing the responses. Also, the feature elicitation process has affected the resulting criteria.
- Reliability: The ratio of the true solutions, assessed to calculate the precision of solutions, is affected by the experts' knowledge.
- External Validity: The age of the companies and also their sizes affect their processes, as the processes are typically tuned during this time. Thus, more weaknesses could be found by using the framework in older/larger companies. In other words, inadequate size and number of samples (data collected throughout interviews and questionnaires) has affected the results of our study.

- Internal Validity: Some specific characteristics of knowledge affect the process for managing it. For example, sharing critical software development knowledge requires considering specific security policies. The framework has not been used to assess the management of such specific kinds of knowledge.

7 Conclusion and Future Work

With the aim of improving KM in SME processes, we have studied the requirements for managing SME process knowledge, with the following contributions: 1) an evaluation framework for identifying the strengths and weaknesses of SME methods in meeting KM requirements and establishing proper SME process knowledge flows, and 2) an improvement model based on the proposed framework. The contributions were validated by evaluating and improving eight SME methods, and also through case studies. The results showed the capability to evaluate and improve SME methods so as to support the KM process.

This research will be continued through defining KM-capability maturity levels for SME methods by categorizing the prescribed solutions within our proposed model, and proposing a novel SME method that provides adequate support for the KM process.

References

[Aharoni and Reinhartz-Berger, 2008] Aharoni, A., Reinhartz-Berger, I.: A Domain Engineering Approach for Situational Method Engineering. In Proc. ER (2008), 455–468.

[Alavi and Leidner, 2001] Alavi, M., Leidner, D. E.: Knowledge Management and Knowledge Management Systems: Conceptual foundations and research issues. MIS Q. 25 (2001), 107–136.

[Ameri and Dutta, 2005] Ameri, F., Dutta, D.: Product Lifecycle Management: Closing the Knowledge Loops. Comput.-Aided Des. Appl. 2 (5) (2005), 577–590.

[Bajec et al., 2007] Bajec, M., Vavpotič, D., Krisper, M.: Practice-Driven Approach for Creating Project-Specific Software Development Methods. Inf. Softw. Technol. 49 (4) (2007), 345–365.

[Bose et al., 2011] Bose, R. P. J. C., Verbeek, E. H. M. W., van der Aalst, W. M. P.: Discovering Hierarchical Process Models Using ProM. In Proc. CAiSE (2011), 33–48.

[Botha et al., 2014] Botha, A., Kourie, D., Snyman, R.: Coping with Continuous Change in the Business Environment. Elsevier (2014).

[Brereton et al., 2008] Brereton, P., Kitchenham, B., Budgen, D., Li, Z.: Using a Protocol Template for Case Study Planning. In Proc. EASE (2008), 1–8.

[Brown, 2011] Brown, J. D.: Likert Items and Scales of Measurement. Statistics 15 (1) (2011), 10–14.

[Clarke and O’Connor, 2012] Clarke, P., O’Connor, R. V.: The Situational Factors That Affect the Software Development Process: Towards a Comprehensive Reference Framework. Inf. Softw. Technol. 54 (5) (2012), 433–447.

[Davison et al., 2013] Davison, R. M., Ou, C. X. J., Martinsons, M. G.: Information Technology to Support Informal Knowledge Sharing. Inf. Syst. J. 23 (1) (2013), 89–109.

[Dehghani and Ramsin, 2015] Dehghani, R., Ramsin, R.: Methodologies for Developing Knowledge Management Systems: An Evaluation Framework. J. Knowl. Manag. 19 (4) (2015), 682–710.

[Dehghani and Ramsin, 2023-1] Dehghani, R., Ramsin, R.: A Knowledge Management-Driven and DevOps-Based Method for Situational Method Engineering. Inf. Technol. Manag. (2023).

[Dehghani and Ramsin, 2023-2] Dehghani, R., Ramsin, R.: Knowledge Management (KM) in Situational Method Engineering (SME): Supplementary File, Version 5. Mendeley Data (2023), https://data.mendeley.com/datasets/gvktzd5ydm/5.

[Di Ciccio et al., 2015] Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-Intensive Processes: Characteristics, Requirements and Analysis. J. Data Semant. 4 (1) (2015), 29–57.

[Durst and Zieba, 2019] Durst, S., Zieba, M.: Mapping Knowledge Risks: Towards a Better Understanding of Knowledge Management. Knowl. Manag. Res. Pract. 17 (1) (2019), 1–13.

[Easterby-Smith and Lyles, 2011] Easterby-Smith, M., Lyles, M. A.: Handbook of Organizational Learning and Knowledge Management. John Wiley & Sons (2011).

[Engels and Sauer, 2010] Engels, G., Sauer, S.: A Meta-Method for Defining Software Engineering Methods. Graph Transformations and Model-Driven Engineering (2010), 411–440.

[Evans et al., 2015] Evans, M., Dalkir, K., Bidian, C.: A Holistic View of the Knowledge Life Cycle: The KMC Model. Electron. J. Knowl. Manag. 12 (1) (2015), 85-97.

[França et al., 2015] França, J., Netto, J., Carvalho, J., Santoro, F., Baião, F., Pimentel, M.: KIPO: The Knowledge-Intensive Process Ontology. Softw. Syst. Model. 14 (3) (2015), 1127–1157.

[García-Borgoñon et al., 2014] García-Borgoñon, L., Barcelona, M. A., García-García, J. A., Alba, M., Escalona, M. J.: Software Process Modeling Languages: A Systematic Literature Review. Inf. Softw. Technol. 56 (2) (2014), 103–116.

[Henderson-Sellers et al., 2014] Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J., Rossi. M.: Situational Method Engineering. Springer (2014).

[Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Q. 28 (1) (2004), 75–105.

[Hoppenbrouwers et al., 2008] Hoppenbrouwers, S. J. B. A., van Bommel, P., Järvinen, A.: Method Engineering as Game Design. In Proc. Int. Conf. on EMMSAD (2008), 97–111.

[Jalali and Wohlin, 2012] Jalali, S., Wohlin, C.: Systematic literature studies: database searches vs. backward snowballing. In Proc. Int. Symp. on ESEM (2012), 29–38.

[Jedlitschka and Pfahl, 2005] Jedlitschka, A., Pfahl, D.: Reporting Guidelines for Controlled Experiments in Software Engineering. In Proc. ESEM (2005), 95–104.

[Kaiya and Saeki, 2006] Kaiya, H., Saeki, M.: Using Domain Ontology as Domain Knowledge for Requirements Elicitation. In Proc. Int. Conf. on Requirements Engineering (2006), 189–198.

[Karni and Levy, 2014] Karni, R., Levy, M.: Tagging Model for Enhancing Knowledge Transfer and Usage during Business Process Execution. In Proc. Int. Conf. on BPM (2014), 429–439.

[Katzenbach and Smith, 2015] Katzenbach, J. R., Smith, D. K.: The Wisdom of Teams: Creating the High-Performance Organization. Harvard Business Review Press (2015).

[Kim et al., 2014] Kim, T. H., Lee, J., Chun, J. U., Benbasat, I.: Understanding the Effect of Knowledge Management Strategies on Knowledge Management Performance: A Contingency Perspective. Inf. Manag. 51 (4) (2015), 398–416.

[Kitchenham et al., 2008] Kitchenham, B., Al-Khilidar, H., Babar, M. A., Berry, M., Cox, K., Keung, J., Kurniawati, F., Staples, M., Zhang, H., Zhu, L.: Evaluating Guidelines for Reporting Empirical Software Engineering Studies. Empir. Softw. Eng. 13 (1) (2008), 97–121.

[Kornyshova et al., 2007] Kornyshova, E., Deneckère, R., Salinesi, C.: Method Chunks Selection by Multicriteria Techniques: An Extension of the Assembly-Based Approach. In Proc. Working Conf. on Method Engineering (2007), 64–78.

[Kuhrmann et al., 2013] Kuhrmann, M., Fernández, D. M., Steenweg, R.: Systematic Software Process Development: Where Do We Stand Today? In Proc. ICSSP (2013), 166–170.

[Leake et al., 2014] Leake, D., Maguitman, A., Reichherzer, T.: Experience-Based Support for Human-Centered Knowledge Modeling. Knowl.-Based Syst. 68 (2014), 77–87.

[Lee et al., 2016] Lee, J., Shiue, Y., Chen, C.: Examining the Impacts of Organizational Culture and Top Management Support of Knowledge Sharing on the Success of Software Process Improvement. Comput. Hum. Behav. 54 (2016), 462–474.

[Li and Liu, 2014] Li, D., Liu, J.: Dynamic Capabilities, Environmental Dynamism, and Competitive Advantage: Evidence from China. J. Bus. Res. 67 (1) (2014), 2793–2799.

[Lin, 2014] Lin, H.: Contextual Factors Affecting Knowledge Management Diffusion in SMEs. Ind. Manag. Data Syst. 114 (9) (2014), 1415–1437.

[Little and Deokar, 2016] Little, T. A., Deokar, A. V.: Understanding Knowledge Creation in the Context of Knowledge-Intensive Business Processes. J. Knowl. Manag. 20 (5) (2016), 858–879.

[Liu and Lin, 2012] Liu, D., Lin, C.: Modeling the Knowledge-Flow View for Collaborative Knowledge Support. Knowl.-Based Syst. 31 (2012), 41–54.

[Marjanovic and Freeze, 2012] Marjanovic, O., Freeze, R.: Knowledge-Intensive Business Process: Deriving a Sustainable Competitive Advantage through Business Process Management and Knowledge Management Integration. Knowl. Process Manag. 19 (4) (2012), 180–188.

[Meihami and Meihami, 2014] Meihami, B., Meihami, H.: Knowledge Management a Way to Gain a Competitive Advantage in Firms (Evidence of Manufacturing Companies). Int. Lett. Soc. Humanist. Sci. 3 (14) (2014), 80–91.

[Mirbel and Ralyté, 2006] Mirbel, I., Ralyté, J.: Situational Method Engineering: Combining Assembly-Based and Roadmap-Driven Approaches. Req. Eng. 11 (1) (2006), 58–78.

[Mundbrod and Reichert, 2017] Mundbrod, N., Reichert, M.: Configurable and Executable Task Structures Supporting Knowledge-Intensive Processes. In Proc. ER (2017), 388–402.

[Nehan and Deneckere, 2007] Nehan, Y., Deneckere, R.: Component-Based Situational Methods. In Proc. Working Conf. on Method Engineering (2007), 161–175.

[Neves et al., 2014] Neves, S. M., Da Silva, C. E. S., Salomon, V. A. P., Da Silva, A. F., Sotomonte, B. E. P.: Risk Management in Software Projects through Knowledge Management Techniques. Int. J. Proj. Manag. 32 (1) (2014), 125–138.

[Osterweil, 2011] Osterweil, L.: Software Processes Are Software Too. In Engineering of Software, Springer (2011), 323–344.

[Palshikar et al., 2011] Palshikar, G. K., Vin, H. M., Saradhi, V. V., Mudassar, M.: Discovering Experts, Experienced Persons and Specialists for IT Infrastructure Support. Serv. Sci. 3 (1) (2011), 1–21.

[Park and Lee, 2014] Park, J., Lee, J.: Knowledge Sharing in Information Systems Development Projects: Explicating the Role of Dependence and Trust. Int. J. Proj. Manag. 32 (2014), 153–165.

[Ramsin, 2006] Ramsin, R.: The Engineering of an Object-Oriented Software Development Methodology. PhD Thesis, University of York (2006).

[Ramsin and Paige, 2008] Ramsin, R., Paige, R. F.: Process-Centered Review of Object Oriented Software Development Methodologies. ACM Comput. Surv. 40 (1) (2008), 1–89.

[Rasnacis and Berzisa, 2017] Rasnacis, A., Berzisa, S.: Method for Adaptation and Implementation of Agile Project Management Methodology. Procedia Comput. Sci. 104 (2017), 43–50.

[Repko and Szostak, 2020] Repko, A. F., Szostak, R.: Interdisciplinary Research: Process and Theory. Sage Publications (2020).

[Riege, 2005] Riege, A.: Three-Dozen Knowledge-Sharing Barriers Managers Must Consider. J. Knowl. Manag. 9 (3) (2005), 18–35.

[Riege and Lindsay, 2006] Riege, A., Lindsay, N.: Knowledge Management in the Public Sector: Stakeholder Partnerships in the Public Policy Development. J. Knowl. Manag. 10 (2006), 24–39.

[Runeson et al., 2012] Runeson, P., Host, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons (2012).

[Seidita et al., 2007] Seidita, V., Cossentino, M., Gaglio, S.: Adapting Passi to Support a Goal Oriented Approach: A situational method engineering experiment. In Proc. EUMAS (2007).

[Serour and Henderson-Sellers, 2004] Serour, M. K., Henderson-Sellers, B.: Introducing Agility: A Case Study of Situational Method Engineering Using the OPEN Process Framework. In Proc. Int. Computer Software and Applications Conference (2004), 50–57.

[Stark, 2020] Stark, J.: Product Lifecycle Management (PLM). In Product Lifecycle Management (Vol. 1), 1–33, Springer (2020).

[Taromirad and Ramsin, 2008] Taromirad, M., Ramsin, R.: An Appraisal of Existing Evaluation Frameworks for Agile Methodologies. In Proc. ECBS (2008), 418–427.

[Team–CMMI Product, 2010] Team–CMMI Product: Improving Processes for Developing Better Products and Services. Version 1.3. Technical report, Carnegie-Mellon University (2010).

[Thomas and Fellmann, 2009] Thomas, O., Fellmann, M.: Semantic Process Modeling–Design and Implementation of an Ontology-Based Representation of Business Processes. Bus. Inf. Syst. Eng. 1 (6) (2009), 438–451.

[Thompson and Walsham, 2004] Thompson, M. P. A., Walsham, G.: Placing Knowledge Management in Context. J. Manag. Stud. 41 (5) (2004), 725–747.

[Wang et al., 2013] Wang, G. A., Jiao, J., Abrahams, A. S., Fan, W., Zhang, Z.: ExpertRank: A Topic-Aware Expert Finding Algorithm for Online Knowledge Communities. Decis. Support Syst. 54 (3) (2013), 1442–1451.

[Wang and Wang, 2016] Wang, Y., Wang, M.: Process Optimization Based on Knowledge Flow in Engineering Change. Procedia CIRP 56 (2016), 406–411.

[Wiewiora et al., 2013] Wiewiora, A., Trigunarsyah, B., Murphy, G., Coffey, V.: Organizational Culture and Willingness to Share Knowledge: A Competing Values Perspective in Australian Context. Int. J. Proj. Manag. 31 (8) (2013), 1163–1174.