On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management

ABSTRACT

Incremental software process improvement deals with the challenges of step-wise process improvement in a time where resources are scarce and many organizations are struggling with the challenges of effective management of software products. Effective knowledge sharing and incremental approaches are essential for improving the success rate of process improvement efforts. During recent years, we have worked on the development of a knowledge management system, the Online Method Engine, that enables incremental, situational process improvement in the field of software product management. This has resulted in an initial system design. In this paper, we describe the findings from seven exploratory case studies on incremental process improvement. The lessons learned during these case studies are used to refine the design of the Online Method Engine.

Keywords: Software Product Management, Software Process Improvement, Knowledge Management, Experience Management, Method Engineering

Introduction

Software process improvement (SPI) has been the subject of many scientific undertakings during the last decades. SPI is not a simple, one-time activity. Changing an organizational structure and its rules and processes has tremendous impact on that organization. During each process improvement effort, several aspects play a role, such as the organizational culture, technical support, and human capabilities (Basili & Green, 1994; Shih & Huang, 2010). Because requirements for SPI efforts are often incomplete, contradictory, ever-changing, and difficult to recognize, and because there is no perfect solution that fulfills all requirements, we can classify SPI as a wicked problem (Churchman, 1967).

Recent SPI literature shows appropriate attention for the wicked nature of SPI. Software engineers have been working with different process improvement models such as CMM (Paulk & Curtis, 1993), SPICE (Dorling, 1993) and the more bottom-up approach Quality Improvement Paradigm (QIP; Basili, 1993). Unfortunately, lack of resources, time pressure, and the difficult nature of change often prevent successful process improvement efforts (Baddoo, 2003).

One key success factor that is mentioned in many studies is the need for iterative and incremental improvement. A systematic literature review revealed that process improvement programs need to be guided in “an iterative and incremental approach (…) that allows a continuous adoption of improvement practices” (Pino, García, & Piattini, 2007). In addition, Sawyer, Sommerville and Viller (1997) state that process improvement should not be seen as a one-step process, but as a sequence of several improvement cycles in which good practice can be introduced in the organization. An advantage of incremental improvement as opposed to revolutionary improvement is that it is a fundamental way to reduce risk on complex improvement projects (Krzanik & Jouni, 2002).

From a knowledge management perspective, incremental improvement has specific advantages as well. Experience suggests that “companies can institutionalize incremental improvement (…) with those doing the work identifying and implementing small changes in product and process” (Davenport, 1993). Furthermore, the introduction of knowledge management in the software development domain led to the new area of experience based process improvement (Sharma, Singh, & Goyal, 2011; Sharma, Singh, & Goyal, 2010). In this approach knowledge that is created during software processes can be captured, stored, disseminated, and reused, so that better quality and productivity can be achieved (Sharma et al., 2010).

Research during the past decade has shown a need for process improvement support that takes the situation of the organization into account, enables incremental implementation of improvements, and that pragmatically leverages existing knowledge and experience (Pino et al., 2007; Sulayman, Urquhart, Mendes, & Seidel, 2012). Unfortunately, although experience management is gaining interest and there is increasing support for building knowledge bases (J. García, Amescua, Sánchez, & Bermón, 2011), there is no proof that current method bases and knowledge infrastructures are effective. Practitioners do not always know exactly what they are looking for, or how to apply a formal method description to the processes of their organization (Niazi, 2011).

During the past years, we have designed an approach to process improvement that addresses the issues of evolutionary improvement, situationality, and knowledge dissemination. This has resulted in the design of the Online Method Engine (OME); a knowledge management tool for incremental SPI. The OME is based upon various other research projects. The design process of the OME can be described in terms of the Systems Development Research Process (SDRP) by Nunamaker Jr., Chen and Purdin (1990)(Figure 1).



Figure 1. Systems Development Research Process

As consistent with the SDRP, the analysis, design and development of the OME are ongoing activities. This means that the conceptual framework, the system architecture and the design of the OME are constantly changing. However, a significant body of research has been published in recent years that forms the conceptual framework on which the development of the OME is based. The main purpose of the next section is to outline this conceptual framework and the resulting initial system architecture. Consequently, we have integrated the related literature discussion with a description of the initial OME design. In the following section, we further analyze the problem and solution space based on the findings from seven exploratory case studies, resulting in an enhanced understanding of the solution space. We describe how these lessons affect the OME design in the section ‘Design Impact’, followed by some conclusions and pointers towards further research.

Towards Incremental Process Improvement Support

The design process of the OME started in 2006 with the design of a knowledge management system for process improvement in the domain of SPM (the Product Software Knowledge Infrastructure or PSKI) described by (van de Weerd, Brinkkemper, Nieuwenhuis, Versendaal, & Bijlsma, 2006). This approach combined an incremental process improvement approach with a method base and the instruments to gather the required data. Using, amongst other, the case study results presented in this paper, the PSKI model has been refined and transformed into the functional architecture of the OMEFigure 1 as depicted in Figure 2. The diagram is shown early in the text in order to structure the remainder of the paper. However, it contains both the original design choices as well as the choices based on the case studies described later on.

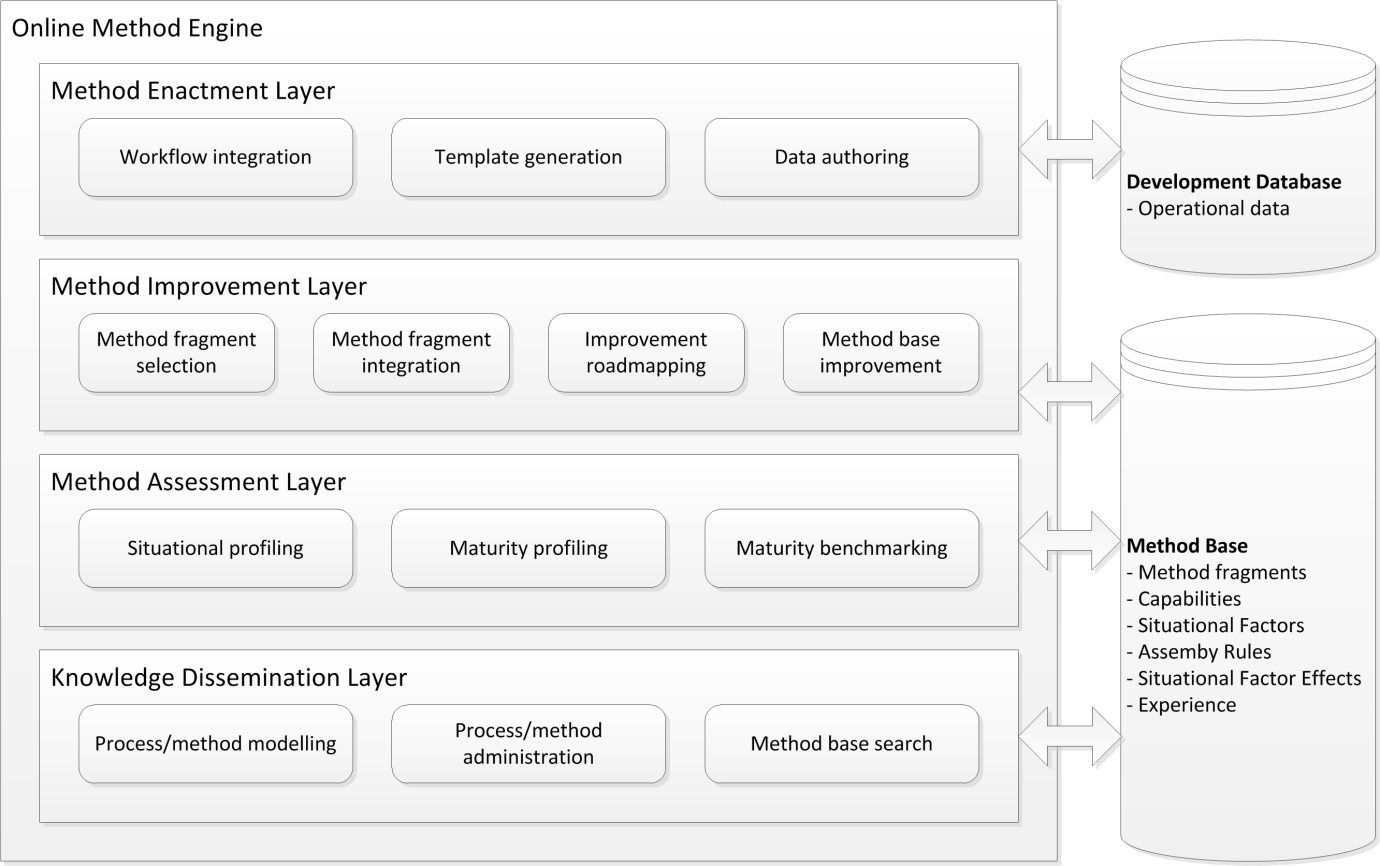


Figure 2. Functional Architecture for the Online Method Engine

Each of the four layers contains a set of functional components, shown by rounded boxes. The layers describe the main functionality related to knowledge dissemination, method assessment, method improvement and method enactment. Each layer builds upon the layer below it.

Knowledge Dissemination

The research described in this paper is grounded in the field of Situational Method Engineering (SME). Brinkkemper (1996) defines the term ’method engineering’ as “the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems”. This is later extended by Harmsen, Brinkkemper, & Oei (1994), who define a situational method as “an information systems development method tuned to the situation of the project at hand”.

Research within the field of SME has focused on structuring available knowledge into small, reusable fragments. The area of SME has been searching for solutions that allow the construction of methods, techniques, and tools based on the specific characteristics and desires of the user. The coverage of the main approaches ranges from the modeling of methods and systems (OMG, 2008; Saeki, 1995) to the situational construction of new ones (Brinkkemper, Saeki, & Harmsen, 1998; Ralyté & Rolland, 2001) and its application to SPI (Bajec, Vavpotič, Furlan, & Krisper, 2007). SME uses the notion of meta-model to describe, alter, and combine methods or parts thereof. During the years, several modularization constructs have been proposed for situational method engineering. Although these constructs have many aspects in common, some essential differences exist. The six main constructs are ’method fragments’, ’method chunks’, ’method components’, ’method services’, ’OPF fragments’ and ’FIPA fragments’. Extensive comparisons of these constructs were performed by Henderson-sellers and Gonzalez-perez (2008), Cossentino, Gaglio, Henderson-Sellers and Seidita (2006), Ågerfalk et al. (2007) and Ralyté, Deneckère and Rolland (2003).

One of the main concepts in most method engineering research is that of the method base. Several researchers (Henderson-Sellers, Simons, & Younessi, 1998; Ralyté, Jeusfeld, Backlund, Kuhn, & Arni-Bloch, 2008; van de Weerd et al., 2006) describe the use of a method base in situational method engineering. Method bases are used to store method fragments in order to make them available for later reuse. Once retrieved from the method base, they can be combined following a set of assembly rules, such as the rules described by Brinkkemper, Saeki and Harmsen (1999). Some method bases have been implemented and put into practice, such as OPF (Henderson-Sellers, 2002) and the CREWS method base (Ralyté, 1999). However, there is no strong scientific proof that practitioners are able to effectively retrieve method fragments from them and use those in their daily work. A prerequisite is that the method user is aware of the exact method fragment that he or she is searching for. In addition, the method user must know what to do with the retrieved method fragments, how to interpret them, and how to implement them in the organization.

The method base concept has been used in the OME design as well, resulting in the knowledge dissemination layer. This layer uses a method base to store information about scientific methods, implemented methods within organizations, experiences with method fragments, and other information relevant to process improvement within the domain of SPM. As such, it forms the backbone of the knowledge management system.

Method Assessment and Improvement

The application of situational method engineering allows for a style of process improvement that is not prescriptive in nature. Prescriptive models are often criticized for being too rigid and 'heavy'. Instead, SME allows for incremental and situational process improvement; incremental in terms of the possibility to improve the process in a series of small steps, and situational in terms of a solution adapted to the situation at hand.

Instead, our research during the last five years has mainly focused on supporting evolutionary process improvement through incremental method evolution (van de Weerd, Brinkkemper, & Versendaal, 2007). During an explorative case study, van de Weerd et al. (2007) defined the atomic types of adaptations that can be made to an existing method, called *method increments*. Insight into these method increments enables focused improvement of software development processes based on localized issues and improvement possibilities. In theory, it allows the linkage of identified local problems to specific changes to the current method. Method increments play a crucial role in our vision on situational and incremental process improvement, and therefore in the design of the OME.



Figure 3. Incremental Process Improvement (Vlaanderen, Brinkkemper, & van de Weerd, 2011)

Based on the the Situational Assessment Method (SAM) by Bekkers, Spruit, van de Weerd, van Vliet and Mahieu (2010), and influenced by process improvement approaches such as the Deming cycle of Plan, Do, Check and Act (Deming, 2000), and the Quality Improvement Paradigm (or experience factory)(Basili, 1993), a high-level approach to incremental method evolution has been proposed by Vlaanderen, van de Weerd and Brinkkemper (2011) (see Figure 3). The approach consists of one major process improvement cycle with one subcycle. The starting point for each process improvement project is an analysis of the current process, based on which a maturity profile can be calculated. The situational context (Bekkers, van de Weerd, Brinkkemper, & Mahieu, 2008) of the organization is used to determine an optimal maturity profile. By calculating the gap between these two, the required process improvement is determined. This process improvement is further detailed by relating it to suitable method fragments that can be combined into a new method that can be implemented to improve the organization’s process.

The method assessment layer of the OME deals with the analysis of methods usingsituational profiling and maturity profiling. These profiles can be used during a process improvement effort, or for direct benchmarking against other organizations within industry.

The method improvement layer contains all functionality related to the translation of process improvement requirements to suitable method increments. This includes the selection of suitable method fragments, the assembly of original and new method fragments into a new method description, and the creation of an improvement roadmap.

Method Enactment

The top layer describes functionality related to method enactment. This includes *workflow integration* to synchronize method descriptions with actual process data, *template generation* to update existing tools such as spreadsheets and requirements databases according to the updated method, and *data authoring* to manage company specific process data.

Method enactment has received some attention in the method engineering literature, but it remains a very complex challenge. The enactment of a situational method requires the integration of multiple tools, models and documents. The ISO/IEC 24744 standard defines enactment as the “act of applying a method[ology] for some particular purpose, typically an endeavor” (ISO/IEC 24744, 2007). The standard also states that enactment is often performed by technical managers. Some researchers have shown that enactment can be facilitated by tools as well (Gonzalez-Perez, 2007; Grundy & Venable, 1996).

Within the OME, the advantages of the underpinning meta-model are used to facilitate deliverable templates and to link and adapt multiple tools. In essence, this layer is strongly related to the concept of Method-as-a-Service (MaaS), described by Rolland (2007), Deneckère, Iacovelli, Kornyshova and Souveyet (2008), and by Guzélian & Cauvet (2007). By adopting a Service-Oriented Architecture (SOA) for method engineering, the authors aim to change method fragments into method services which are implemented as Web services. Deneckère describes how the concept of SOA is adopted in a MOA, a Method-Oriented Architecture. This MOA facilitates a method services registry in which available method services are organized. Unfortunately, the MaaS concept is not thoroughly understood yet.

Technical Architecture

Although the underlying conceptual model is constantly being enhanced, and the system architecture and design have only been partially completed, development of a scientific prototype (cf. Figure 1) has started. The OME is implemented as a hot-pluggable Web Component Bundle (WCB), i.e. a set of components and services based on the concept of an OSGi bundle (Alliance, 2003). The WCB adds a set of custom user interface elements and components to an existing content management system, and allows one to create and maintain a specialist website containing situational method knowledge.

Based on the functional overview of the OME, a technical architecture of the system has been proposed by Vlaanderen, Spruit and Weerd (2012)(see Figure 4). The technical architecture of the OME is based on components and services because, even though the design and implementation are currently guided by the needs of Software Product Management research, the solution should be generic. The solution should be applicable to other domains by replacing, adding or extending components.



Figure 4. Technical Architecture for the Online Method Engine (Vlaanderen et al., 2012)

The current architecture describes the technical realization of the lower two layers of the functional framework. Most components directly reflect the functionality described above. A control panel is defined to support the collection of data, including situational data and method fragments. For the modeling of method fragments, we employ the MetaEdit+ (Tolvanen & Rossi, 2003) meta-modeling tool. This tool enables the definition and usage of domain-specific languages. Communication between MetaEdit+ and the OME environment is structured using XML.

Elaboration of the Online Method Engine Design

In the previous section, we have demonstrated how earlier research on software process improvement and situational method engineering has contributed to the development of a rudimentary design for a knowledge management system for incremental process improvement. This design is subject of constant refinement in order to find a solution that enables us to enhance the effectiveness of process improvement efforts in product software organizations.

Although we are convinced that the main concepts within SPI and SME are generically applicable, research related to the development of the OME is performed within one particular domain. In order to have a clear focus, the research domain is limited to that of Software Product Management (SPM). SPM deals with management of requirements, the definition of releases, and the definition of software products in a context where many internal and external stakeholders are involved (van de Weerd et al., 2006). The domain is described in the form of a competence model by van de Weerd et al. (2006). The domain represents a context where the creation and application of situational methods is very relevant due to relatively low maturity of the field, but where knowledge regarding effective method implementations is scarce.

In addition to the earlier described focus on SPM, small to medium enterprises play an important role within this research. Often facing a low budget for process improvement, these companies have no access to expensive consultants and trainers. Furthermore, their size often allows a large amount of flexibility, and many small to medium enterprises are willing to share detailed information.

In order to further study the relevant concepts, we performed seven exploratory case studies. During these case studies, we tried to determine the current approach to process improvement in terms of process maturity, improvement drivers, and improvement steps (or increments). The lessons learned during these case studies have been used to further define the solution space of the OME. In this section, we outline the main lessons learned from these case studies, and how they affect the OME design.

Case Study Organizations

The case study organizations were carefully selected from a large network of Dutch product software organizations. We have selected organizations that indicated recent process improvement activity related to SPM and/or Scrum. All of the case studies within this research are small-to-medium enterprises. A brief summary of each organization is given below. For the sake of confidentiality, all names have been changed.

*Chatcomp* is a privately-held company, headquartered in Amsterdam. It has local offices in London and San Francisco. The company created an independent, web browser-based instant messaging service in 2003. Within three years, the company grew from 20 employees to over 100 employees. This was accompanied by several extensive changes to the development and product management processes. ChatComp develops two product lines; instant messaging, and real-time messaging for smartphones.

*FacilityComp* is an international software vendor that produces facility management and real estate management software for organizations (Integrated Workplace Management Systems). Founded in 1984, it currently has a customer base of over 1300, which is supported by more than 325 employees. The company's products are marketed through multiple, international subsidiaries, and a worldwide network of partners. For this case, not maturity profile was created.

*SocialComp* is a large social networking site in the Netherlands, focusing mainly on Dutch visitors and members. It was founded in 2004. In 2005, the site reached one million members and eighty million page views from the Netherlands per month. Nowadays, SocialComp has over eleven million user accounts and serves over 5,8 billion page views per month, despite the advent of social networks such as Twitter and Facebook in the Netherlands.

*TimeComp* is a small independent Dutch software company, founded in 1992. The organization provides qualitative software applications and accompanying services to fulfil the need of achieving a higher efficiency from the utilization of human resources. TimeComp currently has two software products in their portfolio: one solution for time resource management, and one solution for printing and copying facilities for organizations. Currently, the company has 25 employees. For this case, only a partial maturity profile was created.

*AgriComp* was established in 1997 in Wageningen, the Netherlands. Initially, the organization focused on plant breeding companies, but they soon expanded their sights to include the entire range of companies within the agribusiness. Until 2003, AgriComp was purely a tailor-made software development company. Currently, it has evolved into a semi-product software company that has two products in its portfolio. Development effort is shared between tailor-made software and the two software products.

*MailComp* is part of the global transportation and distribution industry, and dedicated to providing delivery solutions to its customers. The main organization consists of three pillars, known as ‘mail’, ‘packages’ and ‘e-commerce’. The business unit that was studied in this research is active within the mail division. Due to the nature of its products, MailComp is originally a production-oriented company, with a focus on processing batches of transaction mail. Due to the rise of modern technologies, it is becoming significantly harder for MailComp to maintain its position in the market. Developing and offering software product in addition to their core business is supposed to be the solution for this problem, leading to significant process improvement requirements.

*ServiceComp* was originally founded in 2001 in the Hague, the Netherlands and produces service management products as well as credit management solutions. ServiceComp has years of experience in the development and implementation of IT-solutions for companies in the service-sector. The organization develops standardized, modular web-applications, which are fully customized to the processes, management, users and customers of their clients

Case Study Findings and Effects on the OME Design

Each case study has resulted in a detailed description of the process improvement effort, consisting of a description of the initial process, the improvement steps, the resulting process, and the changes in maturity and context. Case studies are most effective when they have a clear focus. Therefore, we have decided to focus on improvements related to (part of) the SPM activities. In addition, we have analyzed the introduction of the Scrum development process, which affects the SPM processes strongly.

We have summarized the results of these case studies in Table 1T1. For each case, we have identified whether the observed improvements were executed in the light of a process improvement framework (such as CMMI; CMMI Product Team, 2002), what the theme of the improvement was, and whether it was performed in an incremental fashion, or at once. Based on this summary in combination with the more detailed results, we can define several high-level findings that influence the solution space of the OME. These findings have been formulated as generic lessons. However, we are aware that generalization is not possible without further research.

Table 1. Summary of Case Study Improvement Efforts

|  |  |  |  |
| --- | --- | --- | --- |
| Organization | Process Improvement Framework | Improvement | Style |
| ChatComp | None | Introduction of Scrum  Improved Portfolio Management | At once  Incremental |
| FacilityComp | None | Introduction of Scrum for SPM | Incremental |
| SocialComp | None, but part of a larger professionalization effort | Introduction of Scrum  Professionalization of SPM | At Once  Incremental |
| TimeComp | None | Introduction of Scrum  Improved Requirements Engineering | At Once  Incremental |
| AgriComp | None | Professionalization of SPM  Productization | Incremental  Incremental |
| MailComp | None | Professionalization of SPM  Productization | Incremental  Incremental |
| ServiceComp | None, but part of a larger productization effort | Productization | Incremental |

Situational, Experience-Based Process Improvement

The first lesson that we can learn from these results is that none of the studied organizations have performed the identified process improvements in the context of a process improvement framework. Although several frameworks are available that allow a structured and evidence-based approach to selecting and implementing process improvements, they were either not deemed appropriate to the needs of the organization or too heavy to use, or the persons responsible for the process improvement efforts were not aware of their existence or not trained to use them effectively.

Lesson 1: Process improvement frameworks are often deemed inappropriate due to their size or due to their capability requirements.

An important requirement during the design of the OME is therefore that the approach needs to be flexible; simple and quick when organizations want to obtain a quick overview, but thorough and detailed when more resources are available. The organizations that we studied did not use any framework to guide their process improvement efforts, due to various reasons. Although this can be seen as something positive (many frameworks place a heavy burden on the organization), this also implies that the organizations lack guidance and do not have adequate means to measure whether they have reached their goal. The ad-hoc, trial-and-error approach to process improvement indicates that the organizations are not able to effectively leverage experience and knowledge from similar organizations.

The manner in which the organizational processes were changed can be classified as pragmatic. In most cases, a specific problem was identified or experienced. This problem could for instance be an unreliable requirements management process or an inability to cope with the day to day issues of product management. A significant and growing body of methods specific to software product management problems has been proposed during recent years. However, for the selected SPM processes, none of the responsible persons stated that they relied on scientific literature. In most cases, they were unaware of its existence, or they found the literature hard to put into practice.

Lesson 2: Scientific literature is rarely used directly by practitioners, both due to the fact that it is hard to access and that its language is often complex.

In addition, we observed that most organizations scored low on our SPM maturity matrix (Bekkers et al., 2010), mainly with respect to product planning and portfolio management. We have summarized the results of several maturity assessments in Table 2. None of the case companies has used the maturity matrix actively during their process improvement efforts, but the matrix proposes processes that are deemed essential by many industry experts.

Table 2. Maturity Matrices Summary



Table 3. Example Method Fragment Descriptors for Wieger’s Prioritization Technique

|  |  |  |
| --- | --- | --- |
| **Wiegers' Prioritization Matrix** | | |
| *Capabilities* | *Situation* | *Rating* |
| * Internal Stakeholder Involvement * Prioritization Methodology * Customer Involvement * Cost Revenue Consideration * Partner Involvement | * # of requirements < 50 * Partner involvement >= medium | * Ease of use: 8/10 * Satisfaction: 6.5/10 * Additional classifiers |

During the initial phase, it is the process owner that needs to decide what the extent of the analysis will be, i.e. the focus domain. In many cases, it is not required to analyze the entire process. Instead, only a specific part of the process is looked at, and only for that part improvements are provided. For companies that do require a major improvement of their process, this should be a possibility. In those cases, the entire process should be analyzed. This group also contains (new) companies that wish to obtain advice without having a process in place yet, or with a process that is to be abandoned altogether. Although the latter will only very rarely happen, it should be taken into account.

The interviews also suggest that there is a variety of wishes regarding the amount of effort that companies are willing to put in, in order to obtain process improvement advice. In the optimal case, companies are willing to provide complete information regarding their current process, deliverables, and situational factors that describe their environment. This means that their entire process needs to be captured in a way suitable for further elaboration. Also, the situational factors need to be captured in some way, either through a questionnaire or by means of an interview. With all data available, the process improvement advice that can be obtained is the most effective. However, capturing the entire process requires significant work from an expert who is able to employ an appropriate modeling technique.

In many cases, capturing full process information requires too much effort. Therefore, it should be possible to provide a process improvement advice based solely on the situational factors and maturity information. This option implies that the advice does not contain any information on how to implement the advice, but only what should be implemented. If a process owner is willing to provide full information regarding (part of) their process, the process should be modeled by an expert, either internal or external. The resulting model should contain detailed information regarding both the process as well as the deliverables. Therefore, process-deliverable diagrams (PDDs) are a very suitable technique for this purpose. Vlaanderen, van de Weerd, & Brinkkemper (2010) show how PDD's can be used to model an SPM process and to capture the current maturity level of a company's product management process. Based on the choices regarding scope and effort, a questionnaire is generated and performed to gather information regarding the situational context. In the area of SPM, the situational analysis can be performed by conducting a questionnaire with a list of all the relevant situational factors as described by Bekkers et al. (2008).

Process Description Generation

One of the requirements of the SAM is that processes are well-embedded within the organization, and that they are well described in order to avoid loss of know-how when a key stakeholder leaves the organization, and to allow effective knowledge transfer internally. During the case studies, we have found that process descriptions were often not available, and when they were, the descriptions were incomplete or outdated. Some organizations, such as SocialComp explicitly chose not to document their processes for fear of an overly strong focus on formalities. However, especially in fast growing companies such as SocialComp, this can pose a problem in the future.

Lesson 4: Organizational processes are often not documented or they are outdated; either by choice or by lack of resources.

An important function of the OME is therefore the production of relevant method descriptions that help the organization in implementing the suggested improvements and serve as an effective tool for internal communication. The OME should generate complete method descriptions with explanations of all steps, deliverables and roles. These descriptions aid the process owner during the implementation of the process in the company.

The possibilities for embedding or implementing the process advice vary depending on the amount of information that a company has provided. Templates and tools can be generated and/or updated based on the original and the new method description (expressed in the form of PDDs). The possibilities are limited when only maturity information is known, but considerably increase when full method information is given.

If the company's original work documents are available, then they can be updated to reflect the new deliverables within the method. During this step, original data should be maintained while new columns, sections, formulas, etc. are added to the documents. In case deliverables are not available, templates can be generated based on the generated process description. The generated templates should be directly usable within the new process.

Improvement Roadmapping

As can be seen in Table 3Table , improvement of the SPM processes was performed in an incremental fashion in all cases. This fits with theories from the field of method engineering related to method fragments and method increments. The fact that many improvements are implemented step by step can be leveraged and process improvement research should provide fitting solutions, and supporting such steps is one of the main objectives of the OME.

Lesson 5: Process improvements are often implemented incrementally, without a long term process improvement planning.

The first step in the process improvement activity is obtaining an overview of the current situation in terms of implemented capabilities, and situational factors of the business (unit). The current situation constitutes both the currently employed process as well as more generic aspects of the company at hand. The next phase takes the situational factors and the list of implemented capabilities from the first phase as input, after which it determines how the current process could be improved. In the domain of Software Product Management, this phase has already been described by Bekkers and Spruit (2010) in the form of the situational assessment method, but it will be summarized here for the sake of completeness. The need analysis consists of three activities; (1) construction of the current capability profile, (2) calculation of the optimal capability profile, and (3) calculation of an 'areas of improvement' matrix.

The first of these three consists of translating the results from the initial maturity assessment into a form usable for further calculation. The second activity is somewhat more complex. The optimal capability profile is determined by a set of situational factor effects. Several situational indicators have an associated effect. By applying all applicable situational factors effects, an optimal capability profile is obtained that is customized for the current company. The current capability profile and the optimal capability profile are then combined into an Areas of Improvement matrix. This is again a capability matrix, with both previous matrices integrated into it. Between the two matrices, a gap can exist, which we will call the delta. This delta indicates the capabilities that need to be implemented, in order to arrive at the optimal maturity level. An example of such an Areas of Improvement matrix within the Software Product Management domain is shown in Figure 5. The actual delta is the light-grey area, as this depicts the difference between the actual and the optimal maturity level. This set forms the basis for the next phase in the process of method improvement.



Figure 5. Improvement Areas in the Maturity Matrix

Based on the results of the assessment, new method fragments can be selected from the method base. These method fragments can be integrated into the current method, and in case of large improvements, an improvement roadmap can be generated. Steps are needed since solutions will in many cases be too large for implementation in one iteration. An evolutionary approach has more chance of success as it will likely yield a higher acceptance due to smaller, incremental changes.

Although a long term improvement plan was never created in advance, some cases did show a certain theme to which the identified process improvements could be attached. A strong example is that of ServiceComp, which showed a strong productization drive. Its development is shown in Figure 6, which displays the main events and process changes. For each of the stages, we were able to construct a maturity matrix that showed the capabilities relevant to that stage. With a sufficiently filled method base, such experience can be used to create valid improvement roadmaps.

Centralized Knowledge Repository

Another interesting observation is the fact that all companies were very forthcoming in providing us with detailed information regarding their current SPM processes, changes and problems that were experienced in the past, and changes that they plan to make in the future. There is a strong case to make for the possibility that this behavior is specific to small and medium size organization, and that larger organizations are less forthcoming due to stronger internal legislation and a more sensitive competitive position. Still, the fact that the organizations at hand were willing to provide such detailed information enables interesting possibilities in the light of the OME.

*Lesson 6: Many organizations are willing to share detailed information regarding internal processes for the sake of scientific progress and the advancement of the SPM field.*

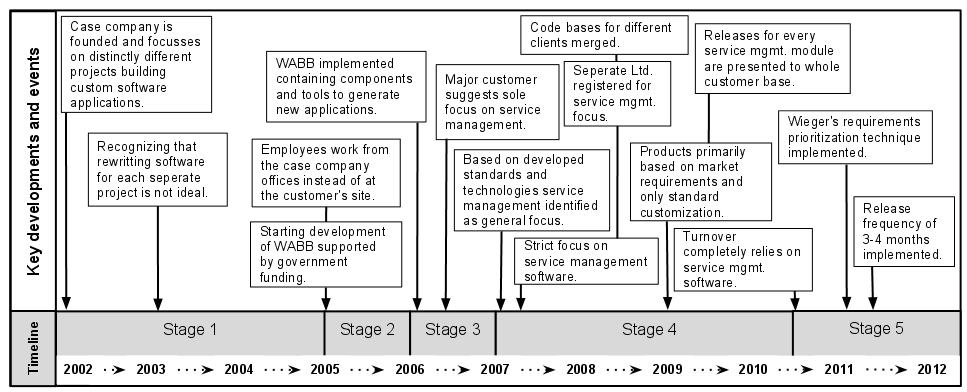


Figure 6. Productization Effort at ServiceComp

This observation directly influences our choice to primarily design the OME as a public platform. By not restricting the environment to individual organizations, it allows benchmarking and knowledge dissemination that leverages the knowledge of a broad range of practitioners. Based on what we have heard during the interviews, we believe that the advantages of increased knowledge availability for SPM outweigh privacy and competitive issues. It does however require a strong focus on data protection and fair use.

To enhance reliability of the organizational data, the employed questionnaires can be replaced by performing an interview. In addition, if a process owner is willing to provide full information regarding (part of) the process, the process should be modeled by an expert, either internal or external. In addition, as the number of available methods and techniques in a given area starts to increase, the difficulty of determining the optimal solution for a given problem becomes more complex. This generally increases the demand on the knowledge of the person or group of persons performing the assessment, or on the quality of the process improvement method.

Due to our focus on small to medium enterprises, we design the OME with the idea that process owners are able to use the system themselves. As the system is highly dependent on an extensive method base, partly created and enhanced by its users, knowledge demands on the user are low. However, as process improvement efforts can be time-consuming, we also foresee the use of the OME by process improvement experts (i.e. consultants, method engineering experts, etc.).

People-Oriented Design

Although processes, capabilities and situational factors form a very solid ground for method fragment selection, we need to take into account that we are dealing with processes in which humans are involved. This means that the resulting process needs to fit with the preferences of the people involved in it. These people need to be able to express these preferences during the selection of alternative method fragments. The results from interviews have indicated that product managers are not always willing to accept suggestions made to them by a machine.

Lesson 7: Process owners are reluctant to trust pure machine-based process improvement suggestions.

Therefore, the approach should allow for differences in the amount of freedom that is provided. Users should be at liberty to select method fragments that differ from the ones proposed. This 'freedom-of-choice' has serious consequences for the OME. In order to make the freedom given to users useful, they need to be provided with a sufficient amount of information for them to base their decision on.

Since every method fragment can be displayed in the form of a PDD, users can use this diagram to form an initial mental image of its implications. All related activities and deliverables are readily available in the method fragment. However, in addition to this, we also identified a need for more sources in the form of experience reports. Experience from people in similar situations is highly valued, and would thus be a valuable addition to the process. Based on all of the sources of information combined, users should be able to make a valid and well-argued choice regarding the method fragments that should be selected, and thus regarding the changes that should be made to the existing process.

Table 4. Linking observations to OME design choices

| DC | Observation | Resulting design choice |
| --- | --- | --- |
| 1 | Process improvement frameworks are often deemed inappropriate due to their size or due to their capability requirements. | Non-prescriptive maturity model with a focus on situationality. |
| 2 | Scientific literature is rarely used directly by practitioners, both due to the fact that it is hard to access and that its language is often complex. | Mixed-model method base consisting of experience data and formal method fragments. |
| 3 | Most organizations score low on the SPM maturity matrix, indicating the lack of activities that are deemed important by many industry experts. | Guidance of the process improvement effort by an industry-proven SPM maturity matrix. |
| 4 | Organizational processes are often not documented or outdated; either by choice or by lack of resources. | Meta-model based process description generation, including activity/deliverable definitions, experience reports and process-deliverable diagrams. |
| 5 | Process improvements are often implemented incrementally, often without a long term process improvement planning. | Combination of Situational Assessment Method and an incremental method engineering approach employing method increments and increment roadmaps. |
| 6 | Many organizations are willing to share detailed information regarding internal processes for the sake of scientific progress and the advancement of the SPM field. | Open access knowledge management environment, enabling benchmarking and effective knowledge dissemination. |
| 7 | Process owners are reluctant to trust pure machine-based process improvement suggestions. | Experience-based process improvement using constant data quality improvement and industry content. |

Design Impact

In this paper, we have described the results from seven case studies that were performed with the aim of exploring the concept of incremental process improvement. From the case studies, we were able to extract several interesting findings that are relevant for the design and development of an online environment that can be used to assess an organization's current processes, create an advice based on this assessment, and align the company's tooling infrastructure with the method improvements.

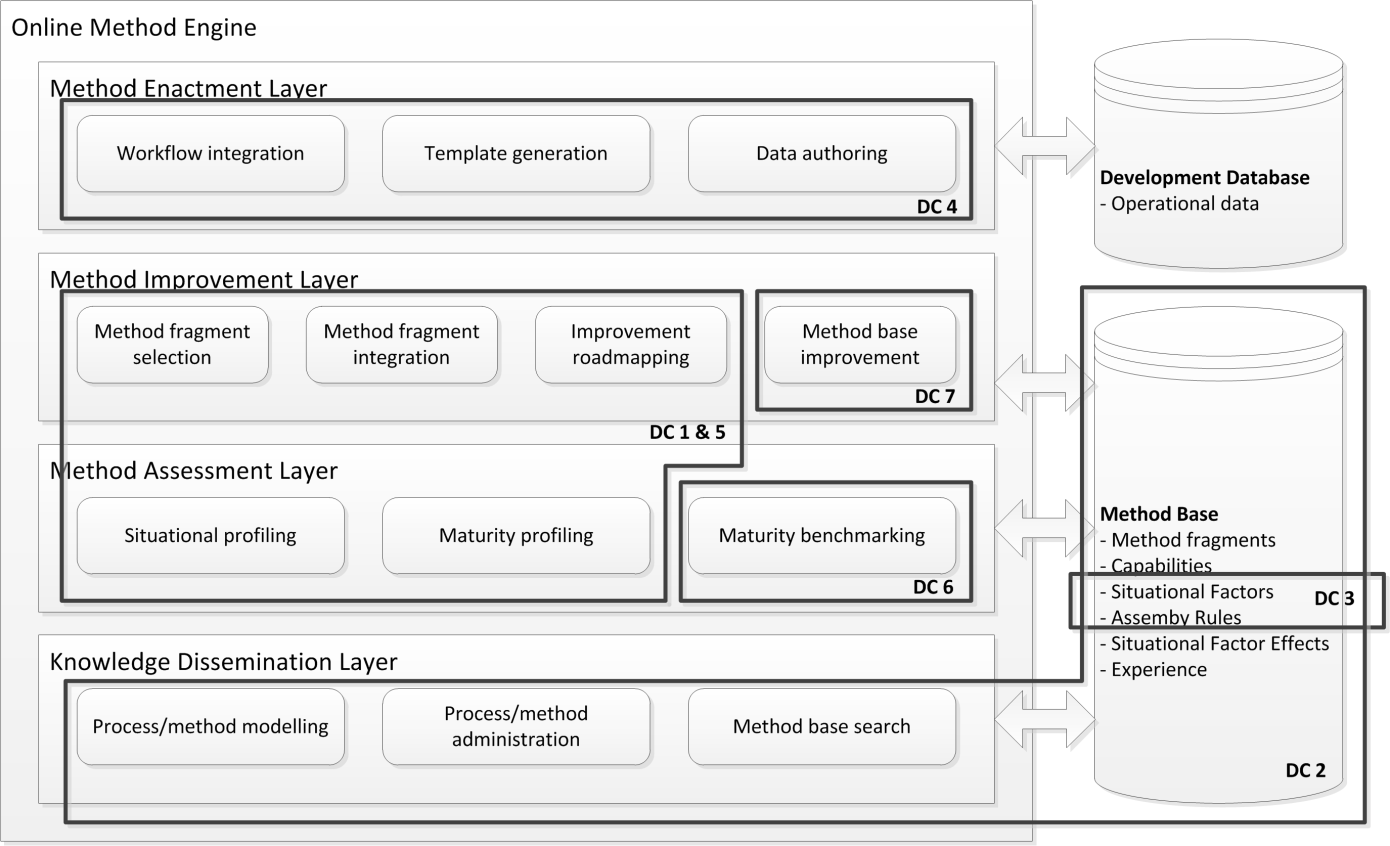


Figure 7. Focus areas of the major design choices

Discussion and Further Research

The knowledge infrastructure that we have described during the discussion of our research approach has evolved into the Online Method Engine (OME) as presented by Vlaanderen, van de Weerd, et al. (2011). The advances in the areas of situational method engineering and incremental process improvement enable us to focus on the *analysis and design of the system* (cf. Figure 1). The findings from the case studies described above have been used to further refine the solution space for the OME.

From this point onwards, each of the areas of the OME needs to be addressed in detail, putting together the architecture and the resulting system. Expertise in several areas will be needed, as each part of the OME has its specific challenges, from linguistic analysis for method assembly to data-optimization for the method base.

Up until this stage, the research effort has mainly been focused on analyzing the current situation and the need, with a focus on the SPM domain. Furthermore, much work has been performed on the underlying meta-modeling techniques that are used throughout the system. This leaves the remaining areas of process alternative selection, improvement roadmap creation, and increment selection and implementation open for future research. An example of a topic that is currently being researched is the creation of an ontology for SPM that should support method description, analysis and construction. Other open issues are the construction of a technique for aligning organizational tooling to method changes and the classification of method fragments.

An important factor that can never be left out during the elaboration is the fact that the purpose of the OME is the improvement of processes. As a consequence, we are always dealing with people that bring habits, experiences, and opinions. This should not be overlooked. Doing so would result in a system that is too rigid, forcing people into ways of working that they will not accept, thereby foregoing the purpose of the system. Unfortunately, the OME that is presented in this paper has not been fully validated yet. As development continues, the user should not be forgotten. At several points in time, the prototype should be validated with practitioners, and corrections should be made according to the results. When done correctly, this will result in a functional online method engineering environment that aids practitioners with the improvement of their SPM processes. We believe that this solution can increase the maturity of the software industry significantly by providing professionals with the right tools to optimize their processes.

Acknowledgements

This paper is an extension of an earlier paper, published in the proceedings of the IFIP WG 8.1 Working Conference on Method Engineering (Vlaanderen et al., 2011). We thank the anonymous reviewers and the editors of the special issue for their suggestions and remarks that led to significant improvement of the paper.

Bibliography

Alliance, O. (2003). *Osgi Service Platform* (3rd ed.). IOS Press.

Baddoo, N. (2003). De-motivators for software process improvement: an analysis of practitioners’ views. *Journal of Systems and Software*, *66*, 23-33.

Bajec, M., Vavpotič, D., Furlan, S., & Krisper, M. (2007). Software process improvement based on the method engineering principles. *Situational Method Engineering: Fundamentals and Experiences* (Vol. 244, pp. 283–297). Springer.

Basili, V. R. (1993). The Experience Factory and its relationship to other Improvement Paradigms. *Proceedings of the European Software Engineering Conference* (pp. 68-83).

Basili, V. R., & Green, S. (1994). Software Evolution at the SEL. *Ieee Software*, *11*(4), 58-66.

Bekkers, W., & Spruit, M. (2010). The Situational Assessment Method Put to the Test: Improvements Based on Case Studies. *Proceedings of the International Workshop on Software Product Management* (pp. 7-16).

Bekkers, W., Spruit, M., van de Weerd, I., van Vliet, R., & Mahieu, A. (2010). A situational assessment method for software product management. In T. Alexander, M. Turpin, & J. van Deventer (Eds.), *Proceedings of the European Conference on Information Systems* (pp. 22-34). Pretoria, South-Africa.

Bekkers, W., van de Weerd, I., Brinkkemper, S., & Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study. *Proceedings of the International Workshop on Software Product Management* (pp. 41-48). Washington, DC, USA: IEEE Computer Society.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275–280. Elsevier.

Brinkkemper, S., Saeki, M., & Harmsen, F. (1998). Assembly Techniques for Method Engineering. *Advanced Information Systems Engineering*, *1413*, 381-400.

Brinkkemper, S., Saeki, M., & Harmsen, F. (1999). Meta-modelling based assembly techniques for situational method engineering. *Information Systems*, *24*(3), 209-228.

CMMI Product Team. (2002). *Capability Maturity Model Integration (CMMI)*. *Engineering* (p. 647). Pittsburgh.

Churchman, C. W. (1967). Guest Editorial: Wicked Problems. *Management Science*, *14*(4).

Cossentino, M., Gaglio, S., Henderson-Sellers, B., & Seidita, V. (2006). A metamodelling-based approach for method fragment comparison. *Proceedings of the 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD06)*.

Davenport, T. H. (1993). *Process innovation: reengineering work through information technology*. New York, NY, USA: Ernst & Young.

Deming, W. E. (2000). *Out of the Crisis* (p. 523). MIT Press.

Deneckère, R., Iacovelli, A., Kornyshova, E., & Souveyet, C. (2008). From Method Fragments to Method Services. *Proceedings of EMMSAD’08*.

Dorling, A. (1993). SPICE: Software process improvement and capability determination. *Software Quality Journal*, *2*(4), 209–224. Springer.

García, J., Amescua, A., Sánchez, M.-I., & Bermón, L. (2011). Design guidelines for software processes knowledge repository development. *Information and Software Technology*, *53*, 834-850.

Gonzalez-Perez, C. (2007). Supporting Situational Method Engineering with ISO / lEe 24744 and the Work Product Pool Approach. *Situational Method Engineering: Fundamentals and Experiences* (Vol. 244, pp. 7-18).

Grundy, J. C., & Venable, J. R. (1996). Towards an Integrated Environment for Method Engineering. *Proceedings of the IFIP WG 8.1 Conference on Method Engineering* (pp. 45-62). Chapman and Hall.

Guzélian, G., & Cauvet, C. (2007). SO2M: Towards a Service-Oriented Approach for Method Engineering. *Proceedings of the international conference IKE’07*.

Harmsen, F., Brinkkemper, S., & Oei, J. L. H. (1994). Situational method engineering for informational system project approaches. *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle* (pp. 169-194). New York, NY, USA: Elsevier Science Inc.

Henderson-Sellers, B. (2002). Process metamodelling and process construction: examples using the OPEN Process Framework (OPF). *Annals of Software Engineering*, *14*(1), 341–362. Springer.

Henderson-Sellers, B., & Gonzalez-Perez, C. (2008). Comparison of Method Chunks and Method Fragments for Situational Method Engineering. *Proceedings of the Australian Conference on Software Engineering* (pp. 479-488). Washington, DC, USA: IEEE Computer Society.

Henderson-Sellers, B., Simons, A., & Younessi, H. (1998). *The OPEN toolbox of techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

ISO/IEC 24744. (2007). *Software Engineering — Metamodel for Development Methodologies* (Vol. 2007, p. 86).

Krzanik, L., & Jouni, S. (2002). Is my software process improvement suitable for incremental deployment? *Proceedings of the International Workshop on Software Technology and Engineering Practice* (pp. 76-87). London, UK.

Niazi, M. (2011). An exploratory study of software process improvement implementation risks. *Journal of Software Maintenance and Evolution: Research and Practice*.

Nunamaker Jr., J. F., Chen, M., & Purdin, T. D. M. (1990). Systems development in information systems research. *Journal of Management Information Systems - Special issue on management support systems*, *7*(3), 631–640. IEEE.

OMG. (2008). *Software & Systems Process Engineering Meta-Model Specification (SPEM)*. *Process Engineering* (p. 236).

Paulk, M. C., & Curtis, B. (1993). The capability maturity model for software. *IEEE Software*, *10*(4), 18-27.

Pino, F. J., García, F., & Piattini, M. (2007). Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, *16*(2), 237-261.

Ralyté, J. (1999). Reusing scenario based approaches in requirement engineering methods: CREWS method base. *International Workshop on Database and Expert Systems Applications (DEXA)* (pp. 305-309). Ieee. doi:10.1109/DEXA.1999.795184

Ralyté, J., & Rolland, C. (2001). An Assembly Process Model for Method Engineering. *International Conference on Advanced Information Systems Engineering, LNCS2068* (pp. 267-283). Springer-Verlag.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. *Advanced Information Systems Engineering* (pp. 1029–1029). Springer.

Ralyté, J., Jeusfeld, M. A., Backlund, P., Kuhn, H., & Arni-Bloch, N. (2008). A knowledge-based approach to manage information systems interoperability. *Information Systems*, *33*(7-8), 754-784.

Rolland, C. (2007). Method Engineering : Achievements,Trends & Challenges. *Keynote presentations of the IFIP WG 8.1 Working Conference on Method Engineering*. Geneva, Switzerland.

Saeki, M. (1995). Object-oriented meta modelling. *Object-Oriented and Entity-Relationship Modeling*, *1021*, 250-259.

Sawyer, P., Sommerville, I., & Viller, S. (1997). Requirements process improvement through the phased introduction of good practice. *Software Process: Improvement and Practice*, *3*(1), 19-34. Citeseer.

Sharma, N., Singh, K., & Goyal, D. (2011). Experience Based Software Process Improvement: Have We Found the Silver Bullet? *Information Intelligence, Systems, Technology and Management*, *141*(2), 71–80. Springer.

Sharma, Neeraj, Singh, K., & Goyal, D. P. (2010). Software Process Improvement through Experience Management : An Empirical Analysis of Critical Success. *Proceedings of the International Conference on Information Systems, Technology and Management* (pp. 386-391). Springer.

Shih, C.-C., & Huang, S.-J. (2010). Exploring the relationship between organizational culture and software process improvement deployment. *Information & Management*, *47*(5-6), 271-281. Elsevier B.V.

Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Software Technology*, *54*(5), 500-479.

Tolvanen, J.-P., & Rossi, M. (2003). MetaEdit+: Defining and Using Domain-Specific Modeling Languages and Code Generators. In R. Crocker & G. L. Steele Jr. (Eds.), *Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA03)*. Anaheim, California, USA: ACM.

Vlaanderen, K., Spruit, S., & Weerd, I. V. D. (2012). *A Technical Architecture for Knowledge Discovery and Method Assessment within the Online Method Engine*. *submitted for publication*.

Vlaanderen, K., van de Weerd, I., & Brinkkemper, S. (2010). Model-Driven Assessment in Software Product Management. In S. Fricker & A. Aurum (Eds.), *International Workshop on Software Product Management (IWSPM)* (pp. 17-25). Sydney, Australia: IEEE CS Digital Library.

Vlaanderen, K., van de Weerd, I., & Brinkkemper, S. (2011). The Online Method Engine: From Process Assessment to Method Execution. In J. Ralyté, I. Mirbel, & R. Deneckère (Eds.), *Proceedings of the IFIP WG8.1 Working Conference on Method Engineering* (pp. 108-122). Paris, France: Springer.

Wiegers, K. (1999). First things first: prioritizing requirements. *Software Development*, *7*(9), 1-6.

van de Weerd, I., Brinkkemper, S., & Versendaal, J. (2007). Concepts for Incremental Method Evolution : Empirical Exploration and Validation in Requirements Management. *International Conference on Advanced Information Systems Engineering* (pp. 469–484). Springer.

van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. In S. Brinkkemper, C. Ebert, & J. Versendaal (Eds.), *Proceedings of the International Workshop on Software Product Management* (pp. 3-12). Minneapolis, USA: IEEE.

Ågerfalk, P. J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S., & Ralyté, J. (2007). Modularization Constructs in Method Engineering: Towards Common Ground? *Situational Method Engineering: Fundamentals and Experiences*, *244*, 359–368. Springer.