Changing Role of SPI – Opportunities and Challenges of Process Modeling

Antero Järvi, Tuomas Mäkilä, and Harri Hakonen

University of Turku,
Department of Information Technology,
FI-20014 University of Turku, Finland
{antero.jarvi, tuomas.makila, harri.hakonen}@utu.fi

Abstract. Software process modeling is gaining acceptance because of the evolving Software Process Engineering Metamodel (SPEM) language. While carrying out empirical process research in software companies in order to model reusable process components with SPEM, we have faced issues that concern Software Process Improvement (SPI) more generally. To understand the general context we have structured these issues into five important aspects of SPI. In this paper we present each aspect through its challenges and opportunities from the process modeling point of view. Consequently, we claim that by overcoming the challenges, process modeling will bring new concrete opportunities for SPI.[1]

1 Introduction

During the last year we have worked on modeling software process frameworks into reusable process components using Software Process Engineering Metamodel (SPEM) process modeling language [1]. The aim of this modeling task has been to identify process content that can be encapsulated as process components and to define guidelines for reusing and tailoring the components for different process contexts. Although the modeling language has the needed expressive power, we constantly ran into situations where we faced many different modeling alternatives, but could not find decisive arguments for choosing between them. This is evidently due to our narrow focus on the modeled process framework as an isolated system; we lacked the software development context where the modeled process framework would be used in. This inspired us to conduct an empirical study on the process needs in different types of software companies, aiming at defining the missing process modeling context. Especially, we concentrated on the variation in processes within the software development companies. We wanted to understand the extent of process variation, and how the companies currently manage to provide process support for different types of projects. It turned out that except for the largest companies, process variation

[1] This paper is based on work done during the ReProCo research project (Sub-project of the E!3320 project) in co-operation with Genestia Group Inc. - Neoxen Systems and Devera Software Development Center.

I. Richardson, P. Runeson, and R. Messnarz (Eds.): EuroSPI 2006, LNCS 4257, pp. 135–146, 2006.

had not been studied. Typically the companies had a single process that was generic enough to fit any project.

Software process modeling has a long research history, but industrial adoption has been slow [2]. The reason for this is twofold. First, SPI itself is complex and evolving issue that still faces many improvement needs [3]. Second, process modeling has a more comprehensive effect on SPI than is generally understood. Process modeling does not simply enhance SPI by making process definition and communication more efficient and increasing process presentation clarity, but it also brings a qualitative change to SPI widening its role in the organization.

Based on our experience in process modeling and the empirical study, we have identified areas that have an important role in realizing the potential of process modeling. This paper is structured along these areas: (i) business, project and process coherence, (ii) process frameworks, (iii) process definition, (iv) SPI cycle, and (v) organization's capability. For each area, we discuss the basic challenges that must be overcome and present opportunities for process modeling technology. The areas are illustrated in Figure 1.

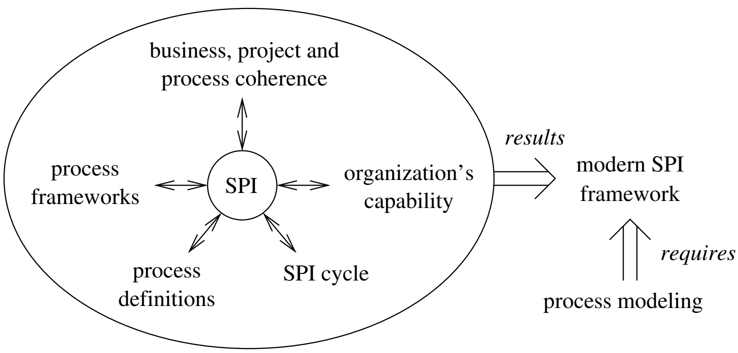

Fig. 1. The areas of concern in realizing the potential of process modeling in a successful SPI function

2 Business, Project and Process Coherence

The fundamental task of the SPI function is to constantly take care of the software processes so that they match the needs of the company's current business objectives. The relevant business goals are company specific and typically involve a mixture of issues of profitability, time-to-market, market share, product strategy, sufficient product and operations quality, and cost efficiency of software development. Also, issues like organizational learning, skills management, core competence and outsourcing management relate to the software development processes. In order to meet the business goals, SPI typically targets process structure and work practices, tooling, quality assurance, compliance to various quality standards or maturity frameworks, software reuse for accumulating long term value, and risk management.

While the fitness of specific processes or methodologies to a certain business context is a common topic in SPI literature, we have found only few cases where the discussion is brought down to practical level where business objectives, described as business factors, are mapped to concrete software process properties. The situation can be clarified by categorizing business factors with respect of stability and volatility, as presented in Figure 2. Stable factors are those that remain unchanged across projects, e.g. organization structure, market situation or product roadmap. Volatile factors vary from project to project including issues like uncertainty of requirements, customer relationship, timeliness requirements, and expected product life span. Traditionally, SPI focuses on stable business factors while volatile project dependent factors are summarized as typical project factors of an average project. This is adequate with similar projects, but fails to provide sufficient process support when the volatile factors vary substantially.

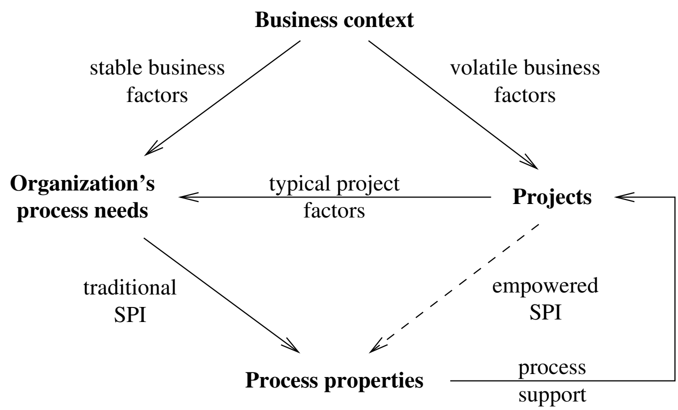

Fig. 2. The stable and volatile business factors map differently to process properties of a project

Coping with the volatile business factors requires empowering SPI with the ability to customize processes for each project based on the project's unique needs. Many popular process frameworks include mechanism for this kind of two-level process tailoring [4,5,6]. However, the frameworks do not explicitly define any business factor taxonomies to guide process tailoring. The required taxonomy consists of (i) business factors and process properties that are categorized and (ii) a mapping from business factors to process properties.

Several business factors have been proposed as the basis for process selection and process customization. We present briefly the approaches of Cockburn and Boehm&Turner. However, neither of these consider the stability of involved factors, instead their work defines common project factor combinations.

Cockburn's Crystal methodology family is adapted to a project in two steps [7]. First a 'methodology type' is selected according to project size and criticality of the developed system. Other factors are taken into account in the second step as priorities that reflect the business objectives that the project faces, e.g.

productivity, repeatability and correctness. The rationale for the first step is that a larger crew needs a more formal methodology and that a more critical system needs more publicly visible correctness in its construction. While this kind of methodology family and selection framework can certainly support high versatility of projects, it is fair to ask how much effort is needed to develop and maintain possibly a few dozen separate processes. Also, the development team's capability to carry out this many methodologies is questionable. However, the factors and priorities are expertly chosen and they certainly capture significant causes of project specific process needs.

Boehm and Turner define two opposite home grounds; one where agile approach is likely to pay off, and another which favors plan-driven methods [8]. They present five critical factors that position a methodology or project with respect to these two home grounds, and also a risk based tailorable method for balancing between them. The positioning factors are system criticality, number of project personnel, skills and capabilities, project dynamism, and organization culture. These factors are used for analyzing the risks of employing agile or plan-driven approach. The process is then tailored to mitigate the risks.

2.1 Challenges

It appears that there is no universal way of choosing which business factors to use as the basis for process variation and the issue is organization and business dependent. For example, the size of the project is generally an important factor, but in some cases other factors like geographical distribution of the development organization can dominate over mere project size.

Thus, the first challenge is to identify the stable and volatile business factors by systematically analyzing the company's business context and projects. Understanding which factors have high priority creates a basis for process design, and is beneficial also for its own sake: The forces that are present can be balanced, risks mitigated and long term business value secured.

The second challenge is supporting the relevant business factors with the software process. This involves selection and adaptation of the process framework, discussed in Section 3, and mapping the business factors to process properties to provide the basis for process tailoring. Mapping the factors to software process properties ties business factors together with development aspects so that they can be resolved together.

2.2 Opportunities

The main opportunity lies in being able to take volatile project factors into account by creating a customized process for a particular project. This is clearly the motivation behind the Crystal methodology family which aims at tailoring a methodology for a project fast enough to get the benefits of customized process before the project is over. Boehm and Turner propose that methods and processes should be built-up, not tailored-down. This should be supported by a repository of 'plug-compatible' process assets that could be quickly adopted, arranged, and used to support specific projects. We share these views and believe that process modeling will provide mechanisms (i) for defining core process structure and content that capture the stable business factors and (ii) for encapsulating the volatile project factors into process components or other reusable or tailorable process assets.

3 Process Frameworks

From the process modeling viewpoint, a process framework describes what must be managed when organizing work, work products, and teams in a given context. The context allows us to have constraints from which, for example, best practices, standardization, and cost-efficiency arise. The most crucial constraints are called dominant assumptions because they define the fundamental characteristics of the process framework. For example, in IBM Rational Unified Process (RUP) it is assumed that Elaboration phase establishes and stabilizes the architecture of a system, and this property is relied on the succeeding Construction phase [5]. In Extreme Programming (XP) process it is assumed that a customer with proper skills and knowledge is constantly available [9]. This is imperative since most of the work in XP relies on instant customer feedback.

Incorporating a new process framework to a company has initiation and managing phases. The initiation of the framework begins with determining and adapting it from the perspective of the company's organization and business context. After this, the process framework is institutionalized to ensure that the organization is able to run it: The skill sets of the company's personnel are supplemented so that the process's practices become organizational process capabilities. This gives concrete means to manage the institutionalized process framework issues, such as project wise tailoring and co-existing frameworks. Figure 3 illustrates the situation where a company initiates projects with agile, RUP, and Microsoft Solution Framework (MSF) [10] process frameworks.

3.1 Challenges

A process framework should be initiated in steps, as described in [11]. In the first step, the process framework is selected and in the second step this company level process is institutionalized. These steps form the two principal challenges of framework adoption.

The first challenge, selecting the framework to fit the company's stable business factors and context, requires understanding the dominant assumptions of the framework under consideration. These assumptions set the limits for the modification of the framework. For example, considering the modifiability, XP is a more specific process than RUP. The applicability of XP is narrow with specific demands, e.g. a single co-located team. Furthermore, the fundamental rules and practices of XP are entwined so that they cannot be altered or removed without in depth analysis of the consequences. On the other hand, the standard RUP is more modifiable but can only be enacted with adaptation. The

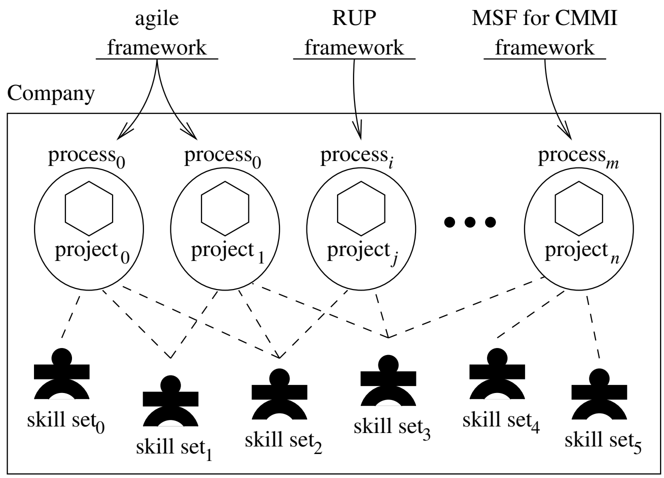

Fig. 3. Initiating process standard frameworks for projects. The projects can run in sequence or in parallel.

goal should be the most specific process framework with respect to the dominant assumptions and modifiability that does not cause foreseeable conflicts. The second challenge in the initiation phase is to institutionalize the selected process framework. This requires that SPI is able to model, manage and utilize personnel skills and organizational capabilities.

A more serious challenge lies in project specific framework adaptation that takes the volatile business factors into account. If the SPI cycle does not keep up with the change rate of the main project factors, it is inevitable that either the work in the project will not conform to the process or the process is used to coerce the work to become inappropriate. This can be an indication of that in terms of dominant assumptions and modifiability a more general or totally different process framework should be used.

Adjusting, adapting, or tailoring an institutionalized process framework is only rarely about including or excluding process elements. For example, it seems unlikely that just removing artifacts, tasks, or roles from a complex framework we get a simpler but still applicable process. If this kind of scalability is possible, it should be defined as a feature in the framework itself. Thus, the SPI's challenge is to cope with the nontrivial management of the company's process frameworks.

A company runs multiple projects in sequence or even in parallel. Because the projects are highly cohesive but rather decoupled, this introduces the continuity problem: How to handle process related know-how, learning, and innovation? It is in the SPI's domain to clarify relationships between the projects that utilize different process frameworks. Process modeling can be used for providing concepts to express explicitly the strategic in-house requirements that affect every process. Process modeling should make SPI more cross-cutting to projects.

All of the preceding challenges call for common conceptualization of the process frameworks. In order to manage multiple institutionalized process frameworks at the same time there must be unifying vocabulary that can describe the processes' similarities and differences. For example, see [12] for various frameworks and [13] for framework attributes taxonomy.

3.2 Opportunities

In a complex problem domain the introduction of common understanding of concepts, relationships, and terminology has often advanced both the research and commercial use. We believe that process modeling will affect SPI similarly, and it will benefit and widen the area where SPI operates successfully. The following process modeling opportunities can contribute to more advanced SPI.

Process modeling separates the definition and use of the process. This means that on one hand, process models can be structured from the perspective of managing large process libraries with efficient tools and practices. On the other hand, the defined processes can be presented in various formats and integrated tightly to project work using tools that are independent of process management.

The opportunity of process management is to achieve specificity and generality at the same time. In practice, process definitions must be structured according the stable and volatile factors using e.g. process components, composition and tailoring mechanisms. This opportunity is materializing rapidly with the appearance of process authoring tools, e.g. Eclipse Process Framework [14] that implements needed process management features.

The opportunity of process usage involves presentation media independence, more interactive ways of presenting the processes, coupling process models to project management for providing automated managerial instruments for planning, monitoring and control, and incorporating processes into integrated development environments for offering a rich process support for the developers.

4 Process Definitions

In every organization a process exists that defines the daily work. In immature or small organizations, the process can be implicitly defined by the culture, tools, document templates and guidelines. This kind of process adapts to the emerging problems in an ad hoc manner with unpredictable results. More mature organizations have explicitly defined processes, forming the basis for continuous process improvement. Process definitions offer a way to analyze the current state of the processes and enable design and communication of process changes.

According to the Capability Maturity Model Integration (CMMI), defined processes are tailored from the organization's process assets [4]. These process assets consist of process descriptions, process element descriptions, life-cycle model descriptions, process tailoring guidelines, and process-related documentation and data. The process elements can be further divided, for instance, into process roles, work products, and applicable procedures. Every process definition should include all of these elements.

There are various business process modeling languages available e.g. traditional flow chart notation, *Business Process Modeling Notation (BPMN)* [15], *Integrated Definition Methods (IDEF)* [16], or *Event-Process Chains (EPC)* [17]. However, software processes have slightly different characteristics than business processes. Where business process modeling is more activity based, the software processes emphasizes the work product flow between the activities. Although there is no de facto standard for software process modeling available, the Software Process Engineering Metamodel (SPEM) [1] is gaining support in the software industry and academic world. With the introduction of SPEM 2.0 in the near future and the ongoing Eclipse Process Framework project [14], we expect to see an accelerated adoption of process modeling.

4.1 Challenges

The SPEM provides a fairly extensive notation for modeling software processes. However, it does not give actual guidelines on how the software processes should be modeled. A challenge is to find out the appropriate accuracy and level of detail of the process definitions. Of course, this is somewhat dependent on the actual purpose of the process modeling. Becker et al. define possible uses for process models, e.g. continuous process management, and identify the required modeling characteristics correspondingly [17].

Second challenge with the process modeling is the creation of models that are equivalent to the actual process. This is not only a question of modeling notation but also about supporting the process implementation with the process definition. It can be argued if too much effort is usually put to the process definition. Better results could be achieved by implementing simple processes and improving them based on the appropriate measurement feedback [18].

Lack of the de facto standard of process modeling causes several challenges to the process definition during the SPI: Tool development becomes slow and expensive, and the absence of the "common language" between process model users also makes maintaining and comparing the process definitions difficult.

All software development stakeholders should be taken into consideration during all parts of the SPI cycle. Fulfilling the different needs of the stakeholders poses a challenge to process definition practices and languages. This challenge is further discussed in Section 6.

4.2 Opportunities

A standard process modeling language that is widely adopted in the software industry and in the research community would yield many benefits. The SPEM could become this kind of de facto process modeling standard. With common process definition notation and guidelines, software process participants could focus on the process definition and modeling itself. In addition the models would be more comparable and interoperable. Reuse of processes within an organization and even between organizations would become possible.

The common modeling notation would be a well-founded start but not sufficient by itself. Process definition conventions should be developed as well. As an example we have proposed a method for increasing re-usability of process definitions by dividing process models into reusable process components [19]. It should be noted that the underlying process framework defines the interfaces and the feasible organization of the process components. Therefore, reuse of process content seems to be restricted to a process framework — process content from different process frameworks are not generally compatible.

5 SPI Cycle

There is a consensus on the basic steps and workflow of SPI cycle in the literature [6, p.46] [20, p.2] [17, p.239] [5, p.253]. The terminology can vary, but basically the continuous software improvement loop always contains the same steps. First the current processes have to be assessed, then the improving changes have to be designed, followed by the implementation of the processes into the organization, and finally the effects of the changes have to be analyzed. The loop is repeated at an appropriate pace.

5.1 Challenges

There are no actual challenges in the general structure of SPI cycle. The consensus on the topic is very firm. The content of the various steps varies greatly, i.e. there are several different methods for assessment of the current processes, designing the changes, implementation of new processes, and analyzing of the results. However, process modeling will affect these SPI steps, as discussed throughout this paper.

5.2 Opportunities

With a proper use of the software process modeling, the SPI cycle could become faster and more efficient. Notably the feedback from process enactment is enhanced. The ideal situation would be that the SPI cycle would not pose any extra overhead to the organization, instead SPI activities would be integrated to other process related activities and to the project management.

6 Organization's Capability

The organizational structure, management practices, culture, responsibility definitions and employees' skills are all constituent elements of SPI. Adopting process modeling techniques raises challenges in all these organizational areas.

6.1 Challenges

The first challenge is to get SPI related responsibilities clear and ensure that SPI's role is understood as a supportive function to operational activities. Different process stakeholders have varying motivators and de-motivators for SPI, reflecting their dissimilar process interests [21,22]. To be successful, SPI should take into account the needs of business management, offer the process as a tool for the project management, and help the development teams to achieve and maintain the needed capabilities. If this is not understood, there is a danger that SPI becomes too detached from the rest of the organization. The process views of these key roles are illustrated in Figure 4. The business management's role is to define the strategic goal of SPI, to provide for the necessary resources, and show managerial commitment to achieve the SPI goals. Process engineers take care of process definition, tailoring, monitoring and improvement. Project management executes projects using the defined processes, and operates as a two way channel between the development teams and the process engineers, relaying feedback and instruction. The development teams' main responsibility is to develop and maintain capabilities required to carry out the process tasks.

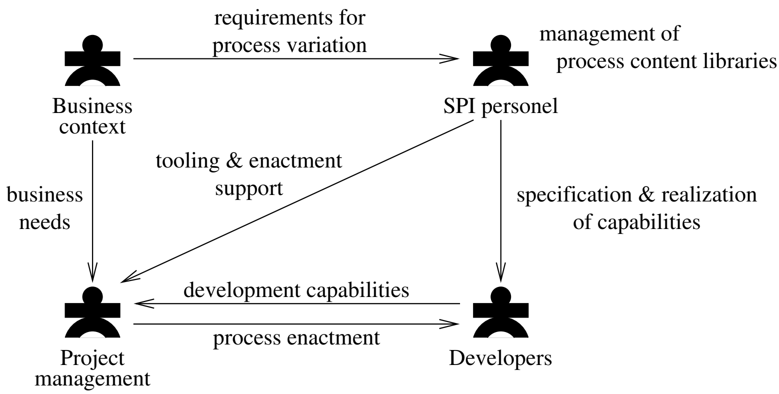

Fig. 4. The roles and relationships of the process stakeholders in a company

The second challenge is to analyze which SPI tasks require special expertise and thus should be allocated to dedicated SPI roles, and which tasks should be carried out by other roles, i.e. project managers, developers or business managers. This can be seen as balancing between centralized and distributed SPI work. The challenge of centralized SPI is its integration to development, whereas the challenge of distributed SPI is to keep SPI coherently working towards the business goals.

Many of the benefits attainable by the use of process modeling are due to speeding up the SPI cycle and defining SPI cycles at various levels, e.g. main SPI cycle for maintaining organization level process libraries, and project level cycles for taking care of project variations. Defined and efficient communication channels both within a cycle and between cycles in different levels are needed.

6.2 Opportunities

Role specific process views can be generated from formally defined process models. This supports the process work of all key roles in the organization. Process modeling technology can be used to explicitly express the balance between centralized and distributed SPI and thus facilitate responsibility allocation in the organization. For example, process components could be used to implement a process variability point linked to a project specific variation. In this way the volatile factors can be tailored locally in the project level without violating organizational level process requirements, such as quality goals. In addition, the above organizational challenges must be considered in realizing any of the opportunities presented in this paper.

7 Discussion

Software process modeling technology is maturing; useful first generation modeling language standard exists, first commercial tools based on the standard language are available, as well as open source solutions. Process modeling is a versatile technology that should not be taken only as a new tool for process engineers. To release the full potential of process modeling, a comprehensive approach on software engineering is needed. In this paper we have discussed how process modeling will affect five important areas of SPI.

The main findings can be summarized as follow. Process modeling enables encapsulating fragments of process content and reusing it to efficiently create customized processes. Identification of business and project factors and taxonomies of process properties are required in order to promote company's business goals with increased process capabilities. Process modeling, specifically a standard modeling language, defines common concepts and terminology, and therefore provides a unifying background for process frameworks. This makes it possible to compare, select from and even deploy several process frameworks for supporting projects with different dominant process assumptions.

The SPI cycle can be accelerated with the use of process modeling technology. More importantly, SPI can be organized as several nested cycles, corresponding to the different levels of process tailoring. Most of the attainable benefits of process modeling require a decentralized SPI function. This requires a clear definition of organizational roles and disseminated process responsibilities.

Both theoretical and applied research on process modeling is clearly needed. As examples, further research should consider existing organizational issues, structure of process libraries, and tooling for all stakeholders. Any research on software process modeling should be tightly connected to practical software development context to ensure pragmatic value. We believe that a comprehensive approach is needed to make process modeling into a mainstream SPI practice in software industry.

References

1. Object Management Group. *Software Process Engineering Metamodel Specification - Version 1.1*, January 5 2005. formal/05-01-06.
2. Alfonso Fuggetta. Software process: A roadmap. In *ICSE - Future of SE Track*, pages 25–34, 2000.
3. Reidar Conradi and Alfonso Fuggetta. Improving software process improvement. *IEEE Software*, 19(4):2–9, 2002.

4. CMMI Product Team. CMMI for systems engineering and software engineering (cmmi-se/sw, v1.1) - staged representation. Technical Report CMU/SEI-2002-TR-002, Software Engineering Institute, Pittsburgh, PA, USA, December 2001.
5. Philippe Kruchten. *The Rational Unified Process: An Introduction (Second Edition)*. Addison-Wesley Professional, March 14 2000.
6. ISO/IEC, Geneva, Switzerland. *ISO/IEC 12207, Information technology - Software life cycle processes*, August 1 1995. ISO/IEC 12207:1995.
7. Alistair Cockburn. Selecting a projects methodology. *IEEE Software*, pages 64–71, July / August 2000.
8. Barry Boehm and Richard Turner. *Balancing Agility and Discipline, A Guide for the Perplexed*. Addison-Wesley, 2003.
9. Don Wells. Extreme programming: A gentle introduction. `http://www.extremeprogramming.org/`, 2006. Accessed on June 22 2006.
10. Msf homepage. `http://msdn.microsoft.com/vstudio/teamsystem/msf/`. Accessed on June 22 2006.
11. Backlund et al. Transfer of development process knowledge through method adaptation and implementation. In *Proceedings of the 11th European Conference on Information Systems (ECIS 2003)*, June 2003.
12. The frameworks quagmire. `http://www.software.org/quagmire/`. Accessed on June 22 2006.
13. Christian Printzell Halvorsen and Reidar Conradi. A taxonomy to compare SPI frameworks. *Lecture Notes in Computer Science*, 2077, 2001.
14. Eclipse process framework project homepage. `http://www.eclipse.org/epf/`. Accessed on June 22 2006.
15. Object Management Group. *Business Process Modeling Notation Specification - Final Adopted Specification*, 2006. dtc/06-02-01.
16. Knowledge Based Systems Inc. Integrated definition methods home page. `http://www.idef.com/`. Accessed on June 22 2006.
17. Becker et al., editor. *Process Management - A Guide for the Design of Business Processes*. Springer, 2003.
18. John Davenport. Don't write another process. *Methods & Tools*, 12(3):2–14, 2004.
19. Antero Järvi and Tuomas Mäkilä. Observations on modeling software processes with SPEM process components. In *Proceedings of The 9th Symposium on Programming Languages and Software Tools*, Tartu, Estonia, 2005.
20. Bob McFeeley. *IDEAL: A Users Guide for Software Process Improvement*. Software Engineering Institute, Pittsburg, PA, February 1996. CMU/SEI-96-HB-001.
21. Nathan Baddoo and Tracy Hall. De-motivators for software process improvement: an analysis of practitioners' views. *The Journal of Systems and Software*, 66:23–33, 2003.
22. Nathan Baddoo and Tracy Hall. Motivators for software process improvement: an analysis of practitioners' views. *The Journal of Systems and Software*, 62:85–96, 2002.