

How Do Artifact Models Help Direct SPI Projects?

Kuhrmann, Marco; Richardson, Ita

Published in:

Proceedings of the 2015 International Conference on Software and System Process

DOI:

10.1145/2785592.2785616

Publication date:

2015

Document version:

Accepted manuscript

Citation for pulished version (APA):

Kuhrmann, M., & Richardson, I. (2015). How Do Artifact Models Help Direct SPI Projects? In *Proceedings of the 2015 International Conference on Software and System Process* (pp. 122-126). Association for Computing Machinery. <https://doi.org/10.1145/2785592.2785616>

Go to publication entry in University of Southern Denmark's Research Portal

Terms of use

This work is brought to you by the University of Southern Denmark.
Unless otherwise specified it has been shared according to the terms for self-archiving.
If no other license is stated, these terms apply:

- You may download this work for personal use only.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying this open access version

If you believe that this document breaches copyright please contact us providing details and we will investigate your claim.
Please direct all enquiries to puresupport@bib.sdu.dk

How do Artifact Models help direct SPI Projects?

Marco Kuhrmann
University of Southern Denmark
Mærsk Mc-Kinney Møller Institute
Odense, Denmark
kuhrmann@mmmi.sdu.dk

Ita Richardson
University of Limerick
Lero - the Irish Software Research Centre
Limerick, Ireland
Ita.Richardson@lero.ie

ABSTRACT

To overcome shortcomings associated with software process improvement (SPI), we previously recommended that process engineers focus on the artifacts to be developed in SPI projects. These artifacts should define desired outcomes, rather than specific methods. During this prior research, we developed a model for Artifact-based Software Process Improvement & Management (ArSPI). We are now carrying out studies to confirm our claims that ArSPI will provide benefits such as quality assurance. In this paper, we report on an experimental setting in which we developed and analyzed a strategy to use artifact models to direct process model improvement. We analyzed a process specification, the realized model, and the generated electronic process guide. We used ArSPI v0.9 as our process model and the Capability Maturity Model Integration (CMMI) as an external reference to provide a set of overall improvement goals. We propose an effective approach to analyze and improve a process model. In addition, the analysis revealed issues with ArSPI realization, which will be corrected in the next major release.

Categories and Subject Descriptors

D.2.9 [Software Engineering Management]: Software process models

General Terms

Management, Experimentation

Keywords

software process improvement, software process management, SPI, SPM, artifact-orientation

1. INTRODUCTION

A variety of software process improvement (SPI) models is presented in literature. Yet, within these papers, the feasibility of these approaches is oftentimes shown by identifying

the realized improvements for the clients and not for other stakeholders. Therefore, as appropriateness is assumed by sufficient customer satisfaction, the quality of the realized model is usually not subject to discussion.

However, SPI is a long-term endeavor in which learning organizations have to deal with evolving processes. For instance, if a software process was initially specified and realized as a process model, it would be unusual for users to implement this model by the letter. Rather, they normally implement it in a situation-specific manner [11]. When it comes to process improvement, process engineers also face the problem of harmonizing originally defined processes with the processes which are actually being used. SPI requires that all parts of a process such as process specification, process realization, and all deployed process assets, e.g., electronic process guides, templates, and collected data and experiences, are considered. Consequently, process engineers must take a broader perspective, as, otherwise, they will encounter the risk of improving a process that is never applied and, eventually, is neglected.

Problem Statement. Since the process model's quality is often out of scope within SPI projects, companies risk the introduction of "technical debt" into their process models. McConnell [9] defines this as "a design or construction approach that's expedient in the short term, but that creates a technical context in which the same work will cost more to do later than it would cost to do now". Furthermore, if processes evolve over time, analyzing the process, locating gaps and the improvement-relevant parts generates extra effort [13]. Hence, process engineers do not only have to improve the process according to new requirements, but also have to deal with previously introduced flaws. This increases the complexity of SPI projects thus increasing cost.

Objectives. To overcome shortcomings associated with SPI, we recommended concentrating on the artifacts to be developed in SPI projects. For this, we developed a model for Artifact-based Software Process Improvement & Management (ArSPI; [8]). By relying on artifact models, this approach explicitly addresses process-model-related quality questions. The goal of this research is to study instruments, which analyze and improve the quality of process models (especially a process model's completeness and consistency) to support structured SPI.

Contribution. In this paper, we present an approach that helps process engineers to determine the quality of a software process and to systematically identify and prioritize improvement candidates. The approach relies on the identi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

fication of process modeling artifacts and their justification with (general) quality attributes, process requirements and external standards, and other complementary material. We report on an exploratory case study, which applied this approach to ArSPI. The case study revealed about 80 technical as well as content-related issues in the studied process model, which were not found in previously conducted quality assurance studies. Initial results suggest that the proposed approach is a useful tool, which can support process engineers in critically reviewing process models and in improving the quality of process models to support long-term process evolution.

Outline. The remainder of the paper is organized as follows: Section 2 summarizes related work. In Sect. 3, we outline the approach, and discuss the results from the case study in Sect. 4. We conclude the paper in Sect. 5.

2. RELATED WORK

In SPI research, the quality of the process model is often not in scope. For example, Unterkalmsteiner et al. [14] present a comprehensive review of evaluation and measurement in SPI. They list 12 evaluation strategies and 14 success indicators. Among 148 publications studied by the first author, no one addresses the quality of the process model as a success factor. Moreover, (standardized) approaches, such as CMMI [2] or ISO/IEC 15504 [4] assess a process by using only content- the structure and the underlying model(s), apparently, do not play a role. We also note that most SPI initiatives and approaches are evaluated by conducting comprehensive case study research with industry, e.g., Raninen et al. [12], to look for those attributes as collected in [14]. The research we present in this paper thus aims at closing a gap in literature by adding another perspective to SPI.

3. APPROACH

The construction procedure of ArSPI [7] motivates the research presented in this paper. We looked for an instrument taking advantage of the artifact-oriented design principle to provide support, which would easily detect issues in the process model and would derive and prioritize improvement measures.

ArSPI Context. Our focus in this paper is the ArSPI artifact *Actual Process*, which is, together with an overall *Vision* and a set of *Changes*, the major input for an improvement project (iteration) [8]. An actual process has a variety of facets: it comprises a *process specification*, a *process realization*, and a *process in use* (Figure 1). A process specification can be represented, for example by standards, norms, and company-wide blueprints. A process realization is often represented by some kind of model¹, for example, through a Software a Systems Process Engineering Metamodel Specification [10] and complementing tools. Finally, the process in use is represented by a variety of artifacts, for example, electronic process guides, documents from past projects, and process user experiences. Calling for a balanced improvement towards user requirements combined with the requirement to consider process model quality improvements, each facet needs to be accounted for in SPI projects as such. Most

¹In this stage of our research, we focus on model-based software processes only.

challenging is the *process in use*. In addition, over time, processes evolve and the actual used process can contradict the -process as originally defined.

Idea and Goals. To analyze a software process for completeness and consistency of the process model and to aid the development and prioritization of improvement requirements, our approach aims to achieve the following goals:

Goal 1: Determine a process model’s quality in terms of completeness and consistency, thus providing a sound and consistent basis for SPI projects.

Goal 2: Determine the current state of a process under investigation in relation to overall improvement goals.

Goal 3: Determine to which extent “cheap” improvements, help to address the improvement goals in general. An example of such improvements would be updating the model’s structure, e.g., regarding consistency.

In a nutshell, the basic idea is to analyze a process for completeness and consistency which can be considered at two levels: 1) the (process) model is complete and consistent based on Bass et al. [1], 2) the resulting description is complete (is the process description missing something) and consistent (is the description free of contradictions). We analyze to which extent “just fixing” completeness and consistency issues helps to address improvement requirements. In addition, we relate found issues to (general) improvement requirements and their prioritization. As we rely on an artifact-based approach, finding all parts of the *Actual Process* is straightforward; to a large extent, this information can be generated from the artifact models.

3.1 Analysis Strategy

The approach to our research is illustrated in Figure 1.

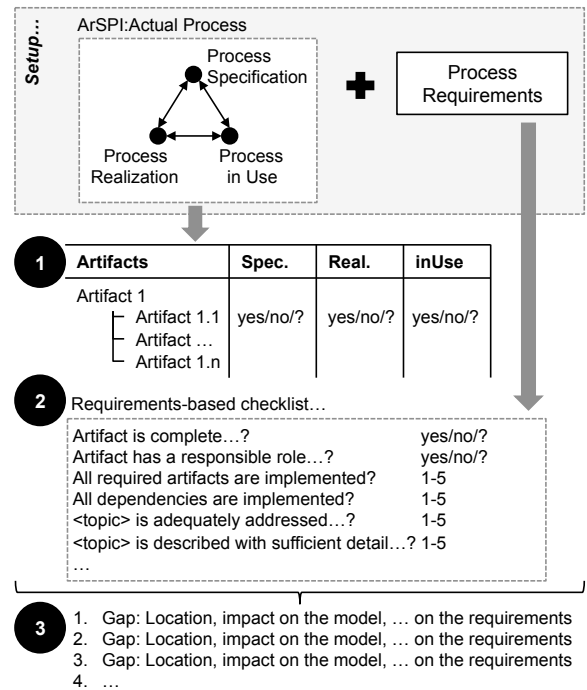


Figure 1: Overview of our approach.

The image shows a screenshot of a CMMI-based checklist titled 'Requirements Management'. The checklist is a table with rows of questions and columns for 'Yes', 'No', 'Doesn't apply', and 'Don't know'. The questions cover various aspects of requirements management, such as system requirements allocation, organizational policy, training, and integration. At the bottom, there is a section for 'Additional comments' and a scale for overall process realization from 1 to 6.

Figure 3: Example CMMI-based checklist for analyzing a process for the purpose of studying how well the Actual Process addresses a particular topic (here CMMI process area Requirements Management).

4.1 Study Setup

To provide an initial application, we opted for the ArSPI model v0.9 [8] for the purpose of defining the requirements for the next version. The study was conducted as a controlled experiment in a Master's Thesis [5]. The student was provided with the specification of ArSPI [6], the process's realization as an EPF model, and the deployment package as HTML web-sites and document templates. CMMI v1.3 [2] was selected as an external reference. The following requirements were defined for the study:

- R1:** The ArSPI model/process consistency and completeness must be ensured.
- R2:** The ArSPI model/process shall be improved to achieve CMMI level 5 or equivalent.
- R3:** Gaps in the ArSPI model/process that hamper the achievement of CMMI level 5 shall be identified, and respective improvement proposals shall be developed and evaluated.

The expected outcomes of the study were an overview of the "hidden" flaws of the model, issues regarding the model's ability to implement high-maturity processes, and a consolidated and prioritized list of (technical) improvement requirements to improve the overall quality of the model.

4.2 Selected Study Results

Figure 2 shows an excerpt from the artifact list generated from the different input. All elements were pair-wise compared to each other to work out the degree of completeness of the different parts in relation to the process specification, and the consistency between the different parts of the actual process. The detailed analysis revealed gaps and issues, which are summarized in Table 1. In the second part of the analysis, CMMI was used to develop a checklist for a content-related analysis (Figure 3). Using this checklist, in a first step, an initial assessment was conducted to generate a baseline. Based on the analysis of the model and the gaps found in step 1, 29 improvement requirements were defined. The requirements were then used to repeat the CMMI-based assessment. This allowed an understanding off which requirements have the most impact on the CMMI-based assessment. From this, technical improvement requirements

were developed and prioritized (goal: minimal effort – maximum impact).

Table 1: Summary of found issues in ArSPI v0.9.

Category	Description
Roles	Missing roles and responsibilities were identified across the whole model, e.g., project- and quality managers, testers, and so forth
Artifacts	16 artifacts from the specification were not properly realized
Artifacts	10 artifact realizations were not in tune with the respective specifications
Activities	37 activities were identified that were neither properly specified nor realized
Activities	17 activities from the specification were not properly realized

An Example. Due to space limitations, we only provide a small example: In the analysis, we found that ArSPI v0.9 was focused on the process engineer, which was the only role specified in the model. Taking the CMMI-based questionnaire, which—among others—asked for responsibilities, we found a major gap resulting in a low rating. In order to address this issue, a requirement was developed, which calls for defining a role model and providing explicit responsibilities. A set of 4 roles (1 from the original model and 3 new roles) was defined and a RACI matrix was developed to provide the requested responsibility assignments. Having defined this requirement, the CMMI-based questionnaire was revisited to analyze the impact of this requirement. Figure 4 exemplary shows the outcomes for a selected CMMI process area and illustrates the dramatic impact the improvement requirement has on the model.

5. CONCLUSION & FUTURE WORK

In this paper, we presented an approach to support process engineers who need to analyze process models. This can emphasize the quality of the process model, especially, the completeness and consistency of the process model under consideration. The approach can determine deviations among the different parts of a process (model), analyze the

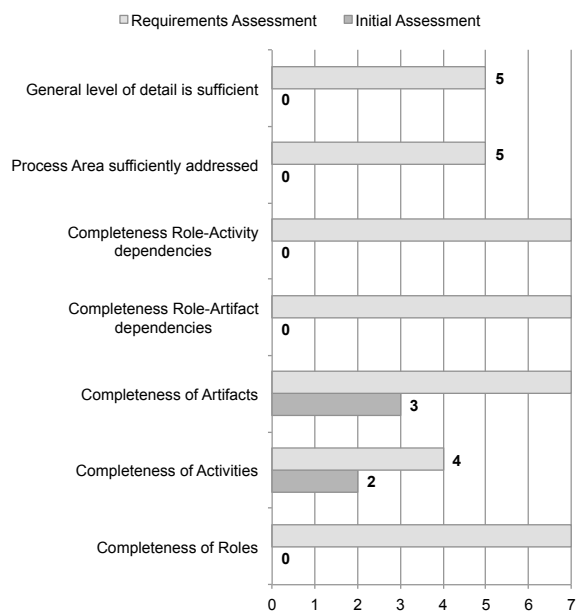


Figure 4: Exemplary results from initial evaluating ArSPI and analyzing the impact of a requirement.

technical requirements for harmonization, and to work out which technical issues found can also help to improve the quality of the process in general. In particular, we aim to reduce the risk of introducing “technical debt” into a process model, which encounters the risk of missing the improvement targets, as improvement goals may address aged processes that are just not used in practice—this would be considered to be specification-based improvement rather than improvement of the actual process.

The approach presented in this paper utilizes the artifact type *Actual Process* as provided by the ArSPI model. The actual process comprises a process specification, a process realization, and process in use components. Using this information, we generate checklists that we use to conduct a pair-wise comparison to find and locate issues regarding completeness and consistency. As second part of the analysis procedure, we use (external) improvement requirements and analyze whether “just fixing” the previously found gaps helps to address the improvement requirements.

Although tentative, we found that the approach using artifact-based improvement procedures to be beneficial. Using artifact models as input simplified the analysis procedures, as all relevant information for the analysis steps could be easily collected and structured. Analysis results were easily linked to tangible objects, allowing us to locate issues in the model and to estimate the impact of potential changes. This also enabled us to provide a prioritized requirements list, to improve the improvement realization procedures and, at the same time, to improve the model’s quality by improving its completeness and consistency.

Limitations and Future Work. Further research is necessary to validate the approach, to demonstrate its feasibility, to improve the approach presented in this paper, and its evaluation in practice. Furthermore, we will make use of the findings gathered so far and implement the requirements found into the next version of ArSPI.

ACKNOWLEDGMENTS

We thank Anastasia Kadnikova for implementing the approach in practice. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie)

6. REFERENCES

- [1] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. The SEI Series in Software Engineering. Addison-Wesley, 3rd edition, 2013.
- [2] CMMI Product Team. CMMI for Development, Version 1.3. Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute, CMU, 2010.
- [3] Eclipse Foundation. Eclipse Process Framework (EPF). Online, <http://www.eclipse.org/epf>, 2010.
- [4] ISO. Software Process Assessment - Part 4: Guidance on use for process improvement and process capability determination. Technical Report ISO/IEC 15504-4:2004, International Organization for Standardization, 2004.
- [5] A. Kadnikova. Assessing and improving an spi method using the arspi framework. Master’s thesis, Technische Universität München, 2014.
- [6] M. Kuhrmann. Arspi: An artifact model for software process improvement and management. Research Report TUM-I1337, TU München, 2013.
- [7] M. Kuhrmann. Crafting a software process improvement approach—a retrospective systematization. *Journal of Software: Evolution and Process*, 27(2):114–145, 2015.
- [8] M. Kuhrmann and S. Beecham. Artifact-based software process improvement and management: A method proposal. In *International Conference on Software and Systems Process*, pages 165–169. ACM Press, 2014.
- [9] S. McConnell. Managing technical debt. Online <http://www.youtube.com/watch?v=1EKvzEyNtbk>, 2011.
- [10] OMG. Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0. Technical report, Object Management Group, 2008.
- [11] D. L. Parnas and P. C. Clements. A Rational Design Process: How And Why To Fake It. *IEEE Transactions on Software Engineering*, 12(2):1–10, 1986.
- [12] A. Raninen, J. J. Ahonen, H.-M. Sihvonon, P. Savolainen, and S. Beecham. LAPPI: A light-weight technique to practical process modeling and improvement target identification. *Journal of Software: Evolution and Process*, 25(9):915–933, 2012.
- [13] M. Sotó and J. Münch. Process model difference analysis for supporting process evolution. In *European System & Software Process Improvement and Innovation*, pages 123–134. Springer, 2006.
- [14] M. Unterkalmsteiner, T. Gorschek, A. Islam, C. K. Cheng, R. Permadi, and R. Feldt. Evaluation and measurement of software process improvement - a systematic literature review. *IEEE Transactions on Software Engineering*, 38(2):398–424, 2012.