OPERATIONALIZATION OF LEAN THINKING THROUGH VALUE STREAM MAPPING WITH SIMULATION AND FLOW

Nauman bin Ali

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2015:05

Department of Software Engineering

Operationalization of Lean Thinking through Value Stream Mapping with Simulation and FLOW

Nauman bin Ali

Blekinge Institute of Technology doctoral dissertation series
No 2015:05

Operationalization of Lean Thinking through Value Stream Mapping with Simulation and FLOW

Nauman bin Ali

Doctoral Dissertation in
Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN

2015 Nauman bin Ali
Department of Software Engineering
Publisher: Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
Printed by Lenanders Grafiska Kalmar, Sweden 2015
ISBN 978-91-7295-302-4
ISSN 1653-2090
urn:nbn:se:bth-00614

Abstract

Background: The continued success of Lean thinking beyond manufacturing has led to an increasing interest to utilize it in software engineering (SE). Value Stream Mapping (VSM) had a pivotal role in the operationalization of Lean thinking. However, this has not been recognized in SE adaptations of Lean. Furthermore, there are two main shortcomings in existing adaptations of VSM for an SE context. First, the assessments for the potential of the proposed improvements are based on idealistic assertions. Second, the current VSM notation and methodology are unable to capture the myriad of significant information flows, which in software development go beyond just the schedule information about the flow of a software artifact through a process.

Objective: This thesis seeks to assess Software Process Simulation Modeling (SPSM) as a solution to the first shortcoming of VSM. In this regard, guidelines to perform simulation-based studies in industry are consolidated, and the usefulness of VSM supported with SPSM is evaluated. To overcome the second shortcoming of VSM, a suitable approach for capturing rich information flows in software development is identified and its usefulness to support VSM is evaluated. Overall, an attempt is made to supplement existing guidelines for conducting VSM to overcome its known shortcomings and support adoption of Lean thinking in SE. The usefulness and scalability of these proposals is evaluated in an industrial setting.

Method: Three literature reviews, one systematic literature review, four industrial case studies, and a case study in an academic context were conducted as part of this research.

Results: Little evidence to substantiate the claims of the usefulness of SPSM was found. Hence, prior to combining it with VSM, we consolidated the guidelines to conduct an SPSM based study and evaluated the use of SPSM in academic and industrial contexts. In education, it was found to be a useful complement to other teaching methods, and in the industry, it triggered useful discussions and was used to challenge practitioners' perceptions about the impact of existing challenges and proposed improvements. The combination of VSM with FLOW (a method and notation to capture information flows, since existing VSM adaptations for SE are insufficient for this purpose) was successful in identifying challenges and improvements related to information needs in the process. Both proposals to support VSM with simulation and FLOW led to identification of waste and improvements (which would not have been possible with conventional VSM), generated more insightful discussions and resulted in more realistic improvements.

Conclusion: This thesis characterizes the context and shows how SPSM was beneficial both in the industrial and academic context. FLOW was found to be a scalable, lightweight supplement to strengthen the information flow analysis in VSM. Through successful industrial application and uptake, this thesis provides evidence of the usefulness of the proposed improvements to the VSM activities.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisors, Prof. Claes Wohlin and Dr. Kai Petersen. Besides their participation and guidance for the thesis, they have been inspirational mentors. As impeccable role models, they have inculcated and enforced a strong work ethic of diligence and integrity. I am also greatly indebted to them for their support throughout this research, and the opportunity and space for both personal and professional growth.

I am thankful to all the practitioners who participated in this research. For their valuable knowledge and time that enabled the studies that are part of this thesis. In particular, I am thankful to Kennet Kjellsson from Ericsson AB for the opportunity to do applied research together.

Colleagues at SERL deserve special praise for their contribution in creating a vibrant and open work environment that is conducive to research. I would especially like to thank Ronald, Bogdan and Michael for all the great times on and after work.

I would also like to thank Sobia and Mr. Tanveer for proof-reading parts of this thesis.

This work would not have been possible without the love, understanding, patience and encouragement from Binish and Yusha. I am also deeply grateful to my parents for their unconditional love, advice, sacrifices and their support in every decision. Last but not least, I am grateful to my brother Umair for encouraging and selflessly supporting me throughout my post graduate studies.

This work was supported by ELLIIT, the Strategic Area for ICT research, funded by the Swedish Government.

Overview of papers

Papers included in this thesis:

Chapter 2. Nauman bin Ali, Kai Petersen and Claes Wohlin, 'A systematic literature review on the industrial use of software process simulation', *Journal of Systems and Software*, Volume 97, November 2014, Pages 65-85, ISSN 0164-1212.

Chapter 3. Nauman bin Ali, Michael Unterkalmsteiner, 'Use and evaluation of simulation for software process education: a case study', In *Proceedings of the European Conference on Software Engineering Education (ECSEE)*, Seeon, Germany, 2014.

Chapter 4. This chapter is an extension of: Nauman bin Ali and Kai Petersen, 'A consolidated process for software process simulation: State of the Art and Industry Experience', In *Proceedings of the 38th IEEE EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, pages 327-336, 2012.
and uses
Nauman bin Ali, Kai Petersen and M. V. Mäntylä. 'Testing highly complex system of systems: An industrial case study', *6th International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2012.

Chapter 5. Nauman bin Ali, Kai Petersen and B. B. N. de França, 'Simulation assisted value stream mapping for software product development: an investigation of two industrial cases', **Submitted** to *Information and Software Technology*, February 2015.

Chapter 6. Nauman bin Ali, Kai Petersen and Kurt Schneider. 'FLOW-assisted value stream mapping in large-scale software product development', **Submitted** to *Journal of Systems and Software*, February 2015.

Chapter 7. Is based on: Kai Petersen and Nauman bin Ali, 'Identifying strategies for study selection in systematic reviews and maps', In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement ESEM*, pages 351-354, 2011.
and
Nauman bin Ali and Kai Petersen. 'Evaluating strategies for study selection in systematic literature studies'. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Turin, Italy, 2014.

Authors' Contribution: Nauman bin Ali is the first author for all the papers except one of the two short papers that constitute Chapter 7. As the lead author, Nauman bin Ali was responsible for leading the research activities, which included: formulation of the research questions, study design, collecting and analyzing data, and most writing activities. In one of the short papers in Chapter 7, Dr. Kai Petersen was the lead author.

Role of advisors: Prof. Claes Wohlin and Dr. Kai Petersen have mainly supported through feedback on research design and reviewing the papers. Furthermore, with their input and in discussion, ideas for studies were defined and further refined.

Role of coauthors: The study reported in Chapter 3 was done by Nauman bin Ali and Michael Unterkalmsteiner (a PhD. student) and the work was equally divided between them.

In Chapter 2, Nauman bin Ali and Dr. Kai Petersen applied the inclusion-exclusion criteria and performed quality evaluation on all the papers independently. This was primarily done to improve the reliability of the secondary study. Prof. Claes Wohlin was extensively involved in this study, his contribution included refinement of research questions, review of the study protocol, discussion on the results and several reviews of the intermediate and final version of the paper with feedback on the formulations.

In the studies reported in Chapters 4, 5 and 6, Dr. Kai Petersen was also involved in the data collection at the company. In one of the two studies reported in Chapter 4, Dr. Mika Mäntylä contributed in the writing of the paper.
Finally, two external researchers collaborated in the capacity of experts in the methodology that was being used in the two studies. In Chapter 5, Breno Bernard Nicolau de França (a PhD. student) acted as an observer reflecting on the way the simulation was used in the study. In Chapter 6, Prof. Kurt Schneider, supported the study with his expertise of FLOW and also contributed to the writing and reviewing of the paper.

Other papers not included in this thesis:

Henry Sitorus, Nauman bin Ali and Richard Torkar. 'Towards innovation measurement in the software industry', *Journal of Systems and Software*, Volume 86, Issue 5, May 2013, Pages 1390-1407.

Kai Petersen and Nauman bin Ali. 'Operationalizing the requirements selection process with study selection procedures from systematic literature reviews', *Proceedings of the 6th Workshop on Requirements Prioritization and Communication, RePriCo'15* co-located with International Conference for Requirements Engineering for Software Quality (REFSQ), Essen Germany, 2015.

Table of Contents

1 Introduction **1**
1.1 Overview . . . 1
1.2 Background and related work . . . 6
1.3 Research gaps and contributions . . . 10
1.4 Overview of chapters . . . 19
1.5 Discussion . . . 26
1.6 Some indications of research impact . . . 33
1.7 Conclusion . . . 34
1.8 Future work . . . 36

2 A systematic literature review on the industrial use of software process simulation **45**
2.1 Introduction . . . 46
2.2 Related work . . . 48
2.3 Research methodology . . . 53
2.4 Characteristics of studies . . . 70
2.5 Systematic literature review results . . . 76
2.6 Discussion . . . 78
2.7 Conclusion . . . 86

3 Use and evaluation of simulation for software process education: a case study **103**
3.1 Introduction . . . 104
3.2 Background and related work . . . 105
3.3 Research design . . . 107
3.4 Results . . . 111
3.5 Analysis and revisiting research questions . . . 116
3.6 Conclusion . . . 118

4 Aggregating software process simulation guidelines: Literature review and industry experience **123**
4.1 Introduction . . . 124
4.2 Related work . . . 125

4.3 Research methodology . . . 127
4.4 Consolidated process for SPSM based on literature sources . . . 133
4.5 Experience of using the consolidated process . . . 143
4.6 Discussion . . . 156
4.7 Conclusion . . . 158

5 Evaluation of simulation assisted value stream mapping: two industrial cases 165
5.1 Introduction . . . 166
5.2 Related work . . . 168
5.3 Framework for simulation assisted VSM (Contribution 1) . . . 169
5.4 Research methodology . . . 174
5.5 Results of the evaluation (Contribution 2) . . . 180
5.6 Discussion . . . 194
5.7 Conclusion . . . 199

6 FLOW assisted value stream mapping in a large-scale software product development 209
6.1 Introduction . . . 210
6.2 Related work . . . 211
6.3 Research methodology . . . 214
6.4 FLOW assisted VSM . . . 218
6.5 Results . . . 221
6.6 Experience and reflections . . . 234
6.7 Follow-up after six-months . . . 239
6.8 Conclusion . . . 240

7 Identifying and evaluating strategies for study selection in systematic literature studies 243
7.1 Introduction . . . 244
7.2 Related work . . . 245
7.3 Research method . . . 245
7.4 Identified strategies (RQ1) . . . 248
7.5 Evaluating study selection strategies (RQ2) . . . 250
7.6 Discussion . . . 257
7.7 Conclusion . . . 259

Chapter 1

Introduction

1.1 Overview

Lean thinking has its origins in the Japanese manufacturing industry and can be traced to work done in the 1950s at Toyota [23]. Womack et al. [80] however are credited to coin the term "Lean production" and for bringing widespread attention to Lean thinking [23]. The continued success of Lean thinking beyond manufacturing [41] [72] has led to an increased interest to adapt and utilize it in software engineering (SE) [58] [73] [22]. Lean thinking is defined as specifying value from the customer's standpoint, mapping and streamlining value-creation activities to deliver it (value stream mapping), developing the ability to conduct these activities without interruption and with predictability (achieving flow), where the development is only triggered by a request (pull-based development), and always striving to perform these activities with ever more effectiveness (continuous improvement) [79]. Womack and Jones [79] propose a step-by-step plan for transformation to Lean as the following: (1) Find a change agent, get the necessary knowledge and competence. (2) Find motivation to introduce change (capitalize on a crisis). (3) Perform Value Stream Mapping (VSM). (4) Picking something important, quickly start with removing waste for immediate returns. From the definition, and the transformation steps the central role and contribution of VSM to operationalize Lean thinking and facilitating organizations in transforming their current way of working is evident [62] [79]. Furthermore, many organizations in various domains have attributed substantial success to the VSM based improvements [41].

However, a review of extant literature on Lean in SE shows that VSM lacks the same recognition in SE [47]. In a systematic literature review, Pernstål et al. [47] found that while VSM is a *"central practice"* in Lean, only two papers report the use of VSM in the SE context out of a total of 38 included papers.

It seems that early adopters of Lean in SE are experiencing the common pitfalls as many manufacturing organizations. As Womack and Jones noted that many organizations skip the most critical step of performing VSM itself and instead rush to eliminate waste from the process [62]. A similar trend is seen in the SE context [47] where only a few studies have reported the use of VSM and the focus has been on applying individual Lean practices and investigating various types of waste [58] in software development. Such well-intended endeavors lead to sub-optimization, as only isolated parts of the overall value stream are improved. Thus, the benefits of Lean transformation in the form of improved quality, reduced cost and shorter time-to-market do not reach the end customer.

VSM is a Lean practice that maps the current product development process (current state map), identifies value adding and non-value adding activities and steps, and helps to create an action plan for achieving an improved future state of the process (future state map) [31] [62] [41] [58] [73]. VSM facilitates to disseminate the current understanding of customer value throughout the entire development organization and helps attain alignment with it. Figure 1.1 illustrates the notation used by Poppendieck and Poppendieck [57] to draw a value stream map. For each activity, the time taken in processing (while a team or a person is actively working on a request) and the total calendar time spent on a request is used to calculate value adding time. Similarly, the waiting times in backlogs between various activities are calculated and captured with this notation.

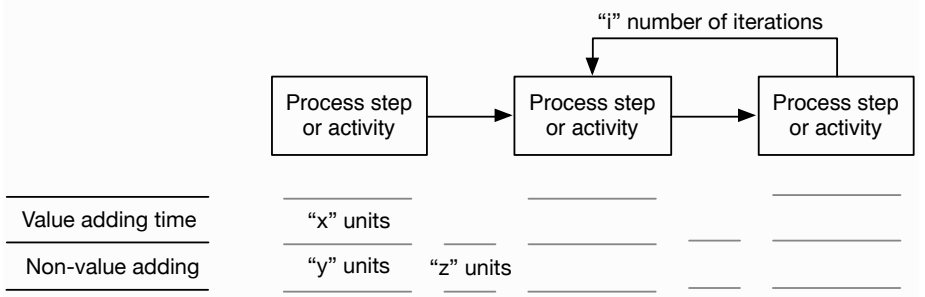


Figure 1.1: VSM notation (based on [57]).

VSM is a practice that *"straddles the gray area between"* concrete practices and analytical principles [22]. It implements several of Lean principles directly (like "optimize the whole", "eliminate waste") and contributes to fulfilling many others (like "continuous improvement", "flow" and "pull-based" development). One of the fundamental principles of Lean is to *"optimize the whole"* [57] and VSM puts this principle to use by taking several measures to take a system-wide perspective, e.g. by considering the end-to-end process and involving multiple stakeholders responsible for various activities in the process [31], both in identification of waste and improvements. It ensures that the focus is on the most significant impediments hindering the organization from delivering value to the customer.

Moreover, revisiting and improving our understanding of customer value and performing VSM to stay aligned with the aim of delivering value to customers efficiently and effectively, provides the necessary means for *"continuous improvement"*, which is another Lean principle [79]. VSM helps to identify waste by providing means to visualize and analyze the current value stream. This helps to operationalize another principle of Lean that is to *"eliminate waste"*. The types of waste (e.g. partially done work, unnecessary features) listed by Poppendieck and Poppendieck [57] should only be considered as a guideline however, it is the VSM activity that will help to see where these manifest in the concrete case of a company's process and which types of waste should be handled with priority.

Figure 1.2, depicts how VSM can systematically operationalize the Lean principles by using them first to guide the analysis of the current value stream, then in identification of waste and lastly in identifying which improvements to implement.

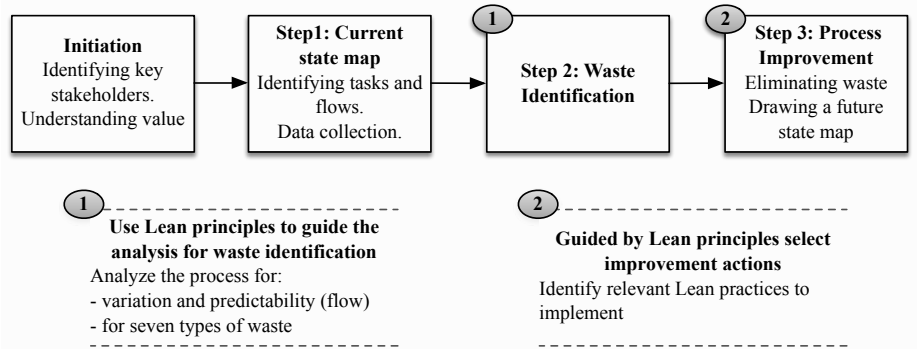


Figure 1.2: Overview of the VSM steps and their role in the operationalization of Lean principles.

Table 1.1 provides an example of how VSM may facilitate the transition to *pull-based* development from the current *push-based* paradigm with large backlogs. This example shows how VSM helps to transition from the abstract principles of Lean (like "achieving flow") to identify and implement concrete practices and actions (like using Kanban and limiting work in progress (WIP) [67]) to realize them. VSM also directly contributes to achieving the Lean principle of "deliver fast" [57] as it explicitly targets waiting times in the process and attempts to reduce the time-to-market.

VSM analyzes both material and information flow [62]. In software product development, an equivalent analysis of material flow will look at the flow of *"work items"*, e.g. a requirement, use case or a user story, through the process (referred to as artifact flow). This has been the focus of current adaptations of VSM outside of manufacturing i.e. for tracing work items through the process, identifying waiting and productive times [41] [31] [43] [58], using a notation similar to the one presented in Figure 1.1. However, for a thorough analysis of the software development process, information flow is equally important to analyze. It is pertinent to capture information needs, knowledge and competence required to carry out the development tasks. The aim is to achieve an information flow that leverages the Lean principle of "pull' such that one process produces only what another process needs [62]. Particularly, in large-scale software development with complex communication structures, it becomes important to focus on the information flow and explicate it. The existing guidelines and notation for VSM [41] [62] do not allow capturing the information flow. As a consequence, we cannot identify value-adding and non-value adding activities required to streamline the process of value creation.

From key literature on Lean [79] [62] and from an application of VSM in an industrial software development context [31], the following two shortcomings were identified in current adaptations of VSM to the SE context. First, the notation used only provides a snapshot of the system and fails to capture the dynamic aspects of the un-

Table 1.1: An example of how VSM helps to identify relevant practices guided by the Lean principle of achieving flow

VSM based analysis	Observations	Improvements
Analyze: - Waiting times and - Variations in the intermediate phases and the process outcomes	- Large waiting times - Long list of backlog items	- Introduce limits on work in progress (WIP) - Use Kanban as a visualization and planning tool - Transition towards pull-based development

derlying processes. This leads to a simplistic analysis to identify bottlenecks. Also, the improvement actions and the target value maps are assessed based on idealistic assertions. These limitations reduce the confidence in the improvement actions identified in VSM, which implies that it is less likely that such improvement actions will be implemented. Second, the current VSM method and notation is unable to capture and represent the myriad of significant information flows, which in software development go beyond just the schedule information about a software artifact's flow through the various phases of a development process.

The thesis attempts to address these shortcomings and emphasizes the use of VSM in the context of Lean software development as a practice that connects existing work on conceptual (e.g. principles of Lean) and tactical levels (concrete practices and processes). For the first shortcoming of VSM, we have proposed the use of Software Process Simulation Modeling (SPSM) as a solution. To achieve this improvement, the usefulness of SPSM was evaluated, and guidelines to perform simulation-based studies in industry were consolidated. This knowledge was used to support VSM with SPSM. To overcome the second shortcoming of VSM, alternatives for capturing rich information flows in software development were explored and a suitable approach was identified to support VSM.

Overall, the thesis attempts to facilitate adoption of Lean thinking in the SE context by supplementing the existing guidelines for conducting VSM. Through successful industrial application, positive evaluation and uptake, the thesis provides evidence of the usefulness and scalability of the improved VSM (including support for simulation and richer information flow modeling) in practice.

Using literature reviews and case study research, the thesis makes the following contributions:

Contribution-1: Recognizes the central role of VSM in operationalization of Lean in the SE context and improves the existing guidelines for conducting VSM.

Contribution-2: Determined the usefulness of SPSM to support VSM in artifact flow analysis and when reasoning about changing the process.

Contribution-3: Determined the utility of FLOW to support VSM to capture, analyze and improve information flows in software development.

Contribution-4: Determined the usefulness of SPSM in applied settings.

Contribution-5: Consolidated the guidelines to apply SPSM in industry.

Contribution-6: Improvement in the guidelines for conducting systematic literature studies by providing means to systematically perform and document study selection related decisions.

The remainder of this chapter is outlined as follows: related work for this thesis is briefly discussed in Section 1.2. Section 1.3 identifies the research gaps and maps them to the contributions of the thesis. It provides the main research question and motivates the choice of the research methods utilized in the thesis. Threats to the validity of the research are also discussed in this section. Section 1.4 provides a summary of each of the chapters in the thesis and Section 1.5 discusses the findings. Section 1.6 presents some indications of the impact of the research in this thesis. Section 1.7 concludes the chapter.

1.2 Background and related work

This thesis proposes and evaluates a solution to an industry relevant problem of effective application of Lean in software development to achieve the improvements in quality, time to market and cost of development as other adopters of Lean. To conduct research on the main topic of interest more effectively, the thesis also proposed and evaluated improvements to guidelines for conducting systematic literature studies, particularly related to study selection criteria and the process.

In this regard, predominantly four main topics of research are used in the thesis, which are: Lean software development and in particular the practice of VSM, SPSM, information flow analysis, and study selection in systematic literature studies.

In the following sections related work on each of these topics is presented briefly. For a detailed discussion of existing research on these topics, see Chapter 2 for SPSM, Chapter 5 for VSM and combination of simulation with VSM, Chapter 6 for information flow analysis in the context of VSM, and lastly Chapter 7 for study selection in secondary literature studies.

1.2.1 VSM

In product development, a context more similar to software development than manufacturing [57], the use of VSM has led to several tangible benefits. McManus [41] reports significant reduction in lead-time and variability of the process and attributed up to 25% of savings in engineering effort to VSM based improvements. The success of VSM outside manufacturing and production and in the context of process improvement in product design/development and engineering made it interesting to explore if it can be leveraged in software development too.

Yang et al. [82] have applied VSM to improve the lead-time of an IT company's R&D process for a hardware component. Mujtaba et al. [43] used VSM as a means

to reduce the lead-time in the product customization process. Kasoju et al. [29] have used VSM to improve the testing process for a company in the automotive industry.

Khurum et al. [31] have extended the definition of waste in the context of software intensive product development. They adapted and applied VSM in the context of large-scale software intensive product development [31].

The shortcomings identified by Khurum et al. [31] are related to the static process models that are currently used in VSM. Given these shortcomings, and the claimed strengths of SPSM over static process modeling (such as its ability to capture uncertainty and dynamism in software development and its perceived low cost of studying a process change [38]), we argue that SPSM can be used to overcome these shortcomings.

1.2.2 SPSM

Software development is a dynamic activity that involves people, tools and processes. The development process is often very complex involving many people, working with various tools and technologies to develop hundreds of requirements in parallel. This complexity [19] makes it challenging to assess the potential impact of the changes proposed for process improvement. Furthermore, changing the development process is a time and resource intensive undertaking. Therefore, the inability to gauge with certainty the likely implications of a change becomes a major barrier in software process improvement.

Software process simulation modeling is proposed as an inexpensive [42] [30] mechanism that attempts to address this challenge by providing proactive means to assess what will happen before actually committing resources for the change [30]. Software process simulation is the numerical evaluation of a mathematical model that imitates the real-world process behavior [7] [38]. Such a computerized model focuses on one or more of the software development, maintenance or evolution processes in particular relevant to the phenomenon of interest [30]. SPSM was first suggested in 1979 by McCall et al. [39] and has been used to address various challenges in the software development ever since, e.g. accurate estimation, planning and risk assessment. The simulation models developed over the years have varied in scope (from parts of the development life-cycle to long-term organizational evolution), purpose (including process improvement, planning, training etc.), and approach (system dynamics, discrete event simulation etc.) [86].

Motivated by the claimed benefits of SPSM in literature that could potentially address the limitations of static models used in VSM, we explored if others have capitalized on this possibility. Based on the results of two systematic reviews on Lean in the SE context [47], [73], we found no studies focusing on the investigation of sim-

ulation assisted VSM for software product development. However, our independent search outside SE yielded encouraging results. Other disciplines have not only combined VSM with simulation, but have done so for largely the similar reasons. Like some other research areas in SE, this highlights the opportunity to "invest in finding and evaluating commonalities and similarities, rather than differences that often appear to be quite artificial" [19]. The following are some of their motivations for combining VSM with simulation (please refer to Chapter 5 for details), which align well with our motivations to supplement VSM with SPSM:

1. VSM alone is time consuming and simulation can assist to speed-up the analysis of the current process, and the derivation and verification of the proposed improvements to the process.
2. It helps to verify the impact of changes and to answer questions that cannot be answered by VSM alone.
3. VSM provides a snapshot, it is unable to detail dynamic behavior. Simulation can help predict the flow and levels and provide more accurate quantitative measures given its ability to handle both deterministic and stochastic inputs.
4. VSM alone cannot capture the complexity in terms of the iterations, overlaps, feedback, rework, uncertainty and stochasticity of the process.
5. Simulation helps to reason about changing the process, and supports consensus building by visualizing dynamic views of the process.

1.2.3 Information flow analysis

None of the reported applications of VSM in the SE context [47] [73] have covered the aspect of information flow analysis. This thesis, is the first to propose and evaluate the combination of VSM with an information flow analysis methodology. Several researchers have investigated dependencies and dynamics of software projects including the impact of information flowing through projects: Pikkarainen et al. [56], for example, investigate communication in agile projects and their impact on building trust. The focus of their work is the impact of agile communication patterns on project success.

Winkler [76] uses the term "information flow" while focusing on dependencies between artifacts. The main interest is in traceability of requirements considering only document-based requirements and information flows. Different ad-hoc illustrations were used to discuss the flow of requirements.

Information flow modeling (FLOW) [68] has been proposed as a systematic method to capture, visualize, and improve situations consisting of a complex network of documented, and undocumented, verbal or informal flow of information. It intentionally uses a very simple notation [64] that can be used on a white-board or a piece of paper to

discuss current and desired situations. The FLOW method uses interviews and begins the analysis by identifying current communication channels and network of information flows [69] [70]. Furthermore, FLOW has been applied to medium-sized groups in several companies where complex networks of information flows were successfully identified, modeled, and discussed with domain experts [68] [83].

Berenbach and Borotto [8] present requirements project metrics. All metrics are based on documented information only. However, information flow within their formal requirements development process could be modeled in FLOW. Some of their metrics could be extended to refer to informal and undocumented information too.

The existing methodology and notation of VSM [41] [62] does not allow to capture the flow of information. As a consequence, we cannot identify value-adding and non-value adding activities required to streamline the process of value creation. Therefore, we needed a systematic, lightweight approach that could supplement VSM, without a lot of overhead. Furthermore, the notation used should be simple and intuitive without additional training of practitioners to understand and analyze the information flows visualized using it. For the solution required for information flow analysis, it was not necessary to create artifacts for documenting information models for the company. Rather, the purpose was to create models that are just detailed enough to support the information flow analysis that is part of VSM. Thus, we needed an information modeling technique that can identify information bottlenecks, unfulfilled information needs, information overload (e.g. on certain employees), and identify mismatch in current storage medium or communication mechanism (e.g. we may be currently relying on face-to-face communication while the product development has been distributed to different geographical sites).

With the above considerations and demonstrated benefits of FLOW in a variety of smaller-scale software development units, we propose and evaluate the combination of VSM with FLOW. This can yield potentially useful results, as both have a focus on capturing and improving information flows. FLOW's systematic approach, and simple graphical notation can compensate the limitation of existing VSM method and notation to purposefully analyze and improve information flows in software development.

1.2.4 Study selection in systematic literature studies

Systematic literature studies [34] [50] are used to explore a variety of topics in software engineering with an aim to answer a research question by conducting an *"exhaustive"* search for relevant literature [34]. Starting with a large set of potentially relevant studies, reviewers rely on several steps of selection, first by reading titles and abstracts, followed by full-text reading for quality assessment [34]. Thus, the reliability of a secondary study is highly dependent on the repeatability of the selection process [78].

Kitchenham and Brereton [33] have aggregated the research on the process of conducting systematic literature studies. They found that the research focusing on the selection of studies has two complementary themes: (1) Using textual analysis tools to assist the selection process e.g. by visualization of citations and content maps to identify clusters of similar studies [16]. Existing work has shown feasibility of the approach and that it can reduce the effort spent on selection [71], but a thorough evaluation of the approach is still required [33]; (2) Making the inclusion/exclusion more systematic. The second theme where strategies for study selection are identified and evaluated is a contribution of the work presented in Chapter 7 of the thesis.

1.3 Research gaps and contributions

The following research gaps were identified and investigated further in this thesis:

Gap-1: Lack of recognition for VSM as the key tool in operationalization of Lean

Gap-2: Very few applications of VSM in the context of software development exist and current adaptations of VSM have two major shortcomings in terms of the use of static process modeling and their inability to capture information flows in software development.

Gap-3: Systematic mapping studies exist, but there is no aggregation of evidence for usefulness of SPSM for the intended purposes.

Gap-4: Lack of guidelines to choose a process for conducting an SPSM based study from a multitude of proposed process descriptions. Furthermore, there is a lack of aggregation of best practices facilitating such a study.

Gap-5: Lack of explicit strategies to guide the selection of articles in systematic secondary studies and their evaluation.

Gap-1 was identified by studying key literature on Lean [79] [80] [62] that identifies VSM as the central practice in Lean transformation. Not finding the same emphasis in the current research on Lean in SE [47] and the apparent disconnect between abstract Lean principles [57] and concrete practices lead to identification of this gap. *Contribution-1:* This thesis recognizes VSM as the connection between the abstract and the concrete when adopting Lean in the SE context (please see Section 1.1 for more details). Furthermore, this theoretical contribution is realized by improving the existing guidelines for conducting VSM.

Gap-2 was identified by reflecting on the limitations in current VSM applications, which were due to the shortcomings of static process models that were being used [31]. Secondly, it was apparent from previous applications of VSM that they have focused

entirely on artifact flow analysis and have overlooked information flow altogether. Furthermore, the current guidelines and notation used in VSM [62] [31] [41] are insufficient to deal with information and communication challenges in software development. *Contribution-2 and Contribution-3:* The shortcomings of using static models in artifact flow analysis were overcome by assisting VSM with simulation. Similarly, to capture the information flows in software development in sufficient detail to enable analysis and improvement, Information flow modeling approach FLOW [69] was identified as a candidate and evaluated to support VSM in a large-scale product development.

Gap-3 was identified by studying the existing literature on the application of simulation in SE. It was observed that there are numerous primary studies applying different simulation techniques for various purposes. However, the existing secondary studies performed poorly on the quality criteria recommended by Kitchenham and Charters for evaluating an SLR [34]. Furthermore, these studies have only scoped the research area and have neither evaluated nor identified the evidence of the usefulness of SPSM. *Contribution-4:* This thesis identifies, aggregates and evaluates the evidence of the usefulness of SPSM from published industrial research. Secondly, the thesis reports results supplementing the existing evidence about usefulness of SPSM in an academic setting.

Gap-4 was identified when the SPSM literature was consulted for a systematic process to guide and support a simulation based study in a company. It was observed that there are a multitude of proposals specializing in the simulation approach used, the experience of modelers or the size of the organization. There are two set of detailed guidelines for individual steps, but both are focused only on SPSM studies using System Dynamics (a technique for simulation modeling) [37] [54]. *Contribution-5:* This thesis reports aggregated good practices to provide practitioners with a process that is based on accumulation of knowledge in SPSM literature. Furthermore, this process was used to develop a system dynamics based training model of the testing process in a company. The experience and reflections on this application were also reported.

Gap-5 was identified during the design of the review protocol for a systematic literature review (SLR) to evaluate the usefulness of SPSM (reported in Chapter 2). It was observed that there were no comprehensive selection guidelines, which was a serious threat to the reliability of an SLR results given that it reduced the repeatability of an SLR. *Contribution-6:* This thesis contributes to the improvement of the SLR guidelines [34] by proposing and evaluating a systematic approach for the selection of articles in systematic secondary studies.

Figure 1.3 illustrates how various contributions address the individual gaps and together achieve the overall aim of this research.

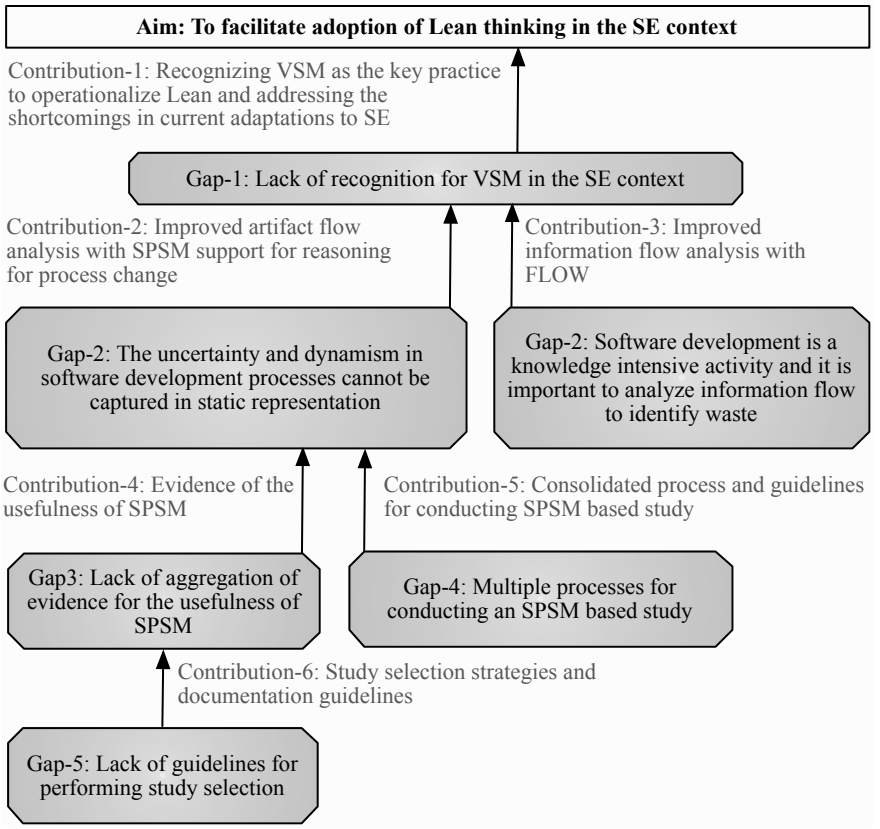


Figure 1.3: Overview of the research gaps addressed in the thesis.

1.3.1 Research questions

The main objective of the thesis is to operationalize Lean thinking and facilitate its adoption in the SE context. On a theoretical level this is achieved by highlighting a disconnect between the abstract concepts and the theory of Lean and the operational or tactical practices. On an applied level the thesis takes a two pronged approach as follows: First it puts forward proposals to combine VSM with simulation and FLOW to overcome the two major shortcomings in current adaptations of VSM. Secondly, it contributes by providing evidence for the usefulness and scalability of these proposed

improvements to VSM in software development. The research question answered in the thesis is:

RQ: How to operationalize Lean thinking in the software engineering context? From a review of relevant literature it was evident that VSM is the key practice that helps organizations in Lean transformation and adopting Lean principles. Thus, our research focus is on using VSM as the tool and answer the following sub-research questions to improve VSM and address the main RQ i.e. operationalize Lean thinking by using VSM.

Sub-RQ-1: How to improve artifact flow analysis in value stream mapping? This is achieved by combining VSM with simulation and assessing which waste and improvements could be identified with this combination. Furthermore, the approach and its outcome were evaluated from practitioners', and an external simulation expert's perspective.

Sub-RQ-2: How to improve information flow analysis in value stream mapping? To answer this question, an appropriate information modeling approach was selected and combined with VSM. The combination is evaluated in the context of a large-scale software product development from practitioners' perspective. The scalability (testing it for such a large product) and usefulness (identifying information flow related challenges and improvement) is also analyzed.

Figure 1.4 illustrates how various chapters complement each other to contribute towards the overall aim of operationalization of Lean thinking in software development through VSM. Chapters 2 to 5, help to answer Sub-RQ-1 and Chapter 6 addresses Sub-RQ-2. The work presented in Chapter 7 improves the guidelines for conducting systematic literature studies and was used to enable research in Chapter 2.

The following is a brief summary of the contribution and connection between various chapters: Chapters 2 and 3 attempt to identify evidence of the usefulness of software process simulation for software engineering. Chapter 2 reports the design and results of a systematic literature review of industrial studies on software process simulation, while Chapter 3 is a primary study where the impact of software process simulation was investigated in an academic setting. Using a literature review, we identified and consolidated the process to conduct a simulation-based study. This process, its application in the case company and the lessons learned are reported in Chapter 4. A case study, undertaken to understand the testing process of our industrial partner to support the simulation based study, is also used in this chapter. Chapter 5 reports the lessons learned from other disciplines that have combined value stream mapping with simulation. It also reports a framework and its evaluation in two industrial cases. Chapter

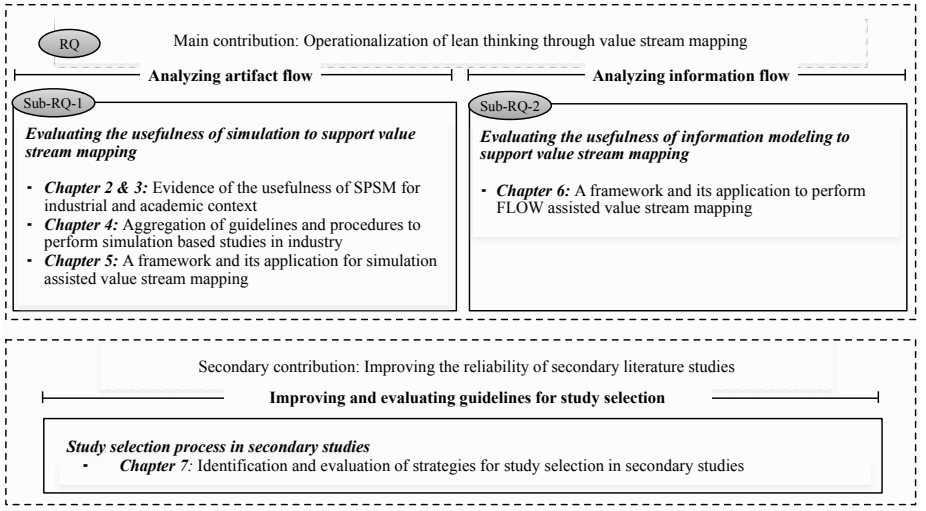


Figure 1.4: Overview of the thesis: research questions, major contributions and the mapping to chapters in the thesis.

6 reports the proposal to support VSM with FLOW and its evaluation in a large-scale product development case. Chapter 7 reports a literature review to identify strategies to reduce bias and resolve disagreements between reviewers in secondary studies (systematic mapping studies and reviews). This chapter also reports the evaluation of identified strategies by utilizing these in a systematic literature review reported in Chapter 2. This work has been used to support the secondary studies conducted in this thesis.

1.3.2 Research method

Research methods most relevant to empirical software engineering [66] include: controlled experiments [77], case study research [63], survey research [18], ethnography [61], and action research [40]. This thesis can be described as *mixed method* research [66]. Different methods were chosen and combined based on their appropriateness to provide the data necessary to answer the research questions in individual studies. An overview of research methods applied in various chapters is provided in Table 1.2. A brief introduction to the research methods applied in the thesis is provided in the following subsections.

Table 1.2: Mapping of the research context and the methods used for research reported in the six chapters of the thesis.

		Chapters					
		2	3	4	5	6	7
Methods	Systematic literature review	✓					
	Literature review			✓	✓		✓
	Case study / Action research		✓	✓	✓	✓	
Context	Industry			✓	✓	✓	
	Academia		✓				✓

Systematic literature review

A systematic literature review aims to exhaustively identify, assess and synthesize evidence related to a specific research question in an unbiased and repeatable manner [34]. A defined review protocol, with an explicit search strategy, inclusion/exclusion criteria, and data extraction forms, guides a systematic review to select and analyze primary studies [34].

This thesis reports one systematic literature review in Chapter 2 that aimed to comprehensively evaluate and assess the claimed usefulness of SPSM for the intended purposes. This review is based on the guidelines by Kitchenham and Charters [34] and uses the study selection process presented in detail in Chapter 7.

Chapters 4, 5 and 7 each report a literature review to identify existing research and get an overview of relevant topics. Hence, thorough quality assessment and synthesis were not required. The reviews were still done with a defined search strategy, an explicit selection process and a described data extraction and analysis process. However, we do not consider these systematic literature reviews as the purpose was not to aggregate and evaluate the evidence reported in existing research [50].

Case study

Given the nature of software development which is very context dependent [51], it is difficult to study a phenomenon of interest in isolation. Thus, case studies are a highly relevant research method as they investigate a contemporary phenomenon within its natural context [63]. It has a high degree of realism, but that comes at the expense of the level of control that one may achieve in controlled experiments. The credibility of the results is improved by triangulation of data sources, observers, methods and

theories [63]. Case studies use a flexible design approach which allows change of design based on data gathered e.g. more data can be gathered if the data collected is insufficient for analysis. It may involve various iterations of the following steps [63]: case study design, preparation of data collection, collecting evidence, data analysis, and reporting.

Chapters 3, 4, 5 and 6 all report case study research conducted as part of this thesis. All of these studies involved the introduction of an intervention to improve the problem situation and simultaneously contribute to scientific knowledge. This bears similarity to action research, where a researcher actively participates in planning, implementing, monitoring and evaluating the impact of the introduced change [40]. A distinction between action research and case study research with the purpose of improving certain aspect of the phenomenon being studied [63] is not always straight forward. For example, in Chapter 4 case study and literature review were used within the framework of action research methodology [49]. A practical problem was identified in the company, we conducted a case study to understand the problem context and conducted a literature review to understand the state-of-the-art. Similarly, taking on the role of practitioners we developed the simulation model in the company and evaluated the process derived from literature.

Chapter 3 reports a case study done in an academic context (in an active course) to supplement existing evidence since the use of simulation has mostly been evaluated in purely experimental setting, whereas Chapters 4, 5 and 6 report studies conducted in industrial settings. These chapters provide a detailed account of the context of these studies.

1.3.3 Validity threats and limitations

Different classifications exist for validity threats of empirical research, e.g. for experimentation [77] and for case studies [63]. The threats to the validity of findings in the thesis are discussed using the classification by Runeson and Höst [63]. It was used as most of the studies in this thesis have used case study research.

Reliability

The validity threats to the reliability of a study are related to the repeatability of a study i.e. how dependent are the research results on the researchers who conducted it [63].

This threat was minimized by involving multiple researchers in design and execution of the studies. For example, for the systematic review reported in Chapter 2, it meant review of the study protocol by two additional reviewers, two reviewers independently applying selection criteria and performing quality assessment on all the

potential primary studies. This was done to achieve consistency in the application of the criteria and to minimize the threat of misunderstanding by either of the reviewers. To minimize reviewers bias and dependence of review results on personal judgments, explicit criteria and procedure were used. These measures along with the detailed documentation of the process also increase the repeatability of the review. In Chapter 4, only one reviewer did the selection of studies, data extraction from the selected primary studies and the analysis of the extracted guidelines. This means that there is a risk of bias, and a threat to the validity of results exists. However, this threat of bias was reduced by having explicit objective criteria to guide selection and data analysis (coding of individual guidelines) was reviewed by a second researcher.

Similarly, in all case studies, explicit case study protocols with documented steps for data collection and analysis were used. Generally, two researchers did data collection and analysis. Where it was not possible or practical to duplicate the effort for analysis (e.g. due to time constraints introduced due to practitioners availability for studies reported in Chapters 5 and 6), additional measures were undertaken to reduce bias and ensure consistent analysis of data. For example, with the review of analysis and results by a second researcher and then through feedback from the practitioners. Practitioners' feedback on intermediate results and observations was collected throughout the studies.

Furthermore, involvement of external researchers as observers in studies reported in Chapters 5 and 6 also helped improve the reliability of results by providing an additional means of observer triangulation.

Internal validity

The factors that the researcher is unaware of or cannot control the extent of their effect, limit the internal validity of studies investigating a causal relation [63].

For literature reviews, we tried to minimize the risk of overlooking relevant literature by taking measures such as: searching in venues atypical of computer science and software engineering e.g. using business and management literature as well, by using guidelines presented in Chapter 7 for selection of articles that aim to reduce bias in selection and document the selection decisions, and by supplementing protocol driven database search with backward and forward snowball sampling [75]. There is some empirical evidence that snowballing provides improved results when compared to database search [28] [21].

For case studies, involvement of multiple researchers in interviews, workshops and retrospective meetings allowed for triangulation of notes and observations acquired from these meetings by individual researchers. This reduced the threat of misinterpretation of feedback by the researchers. In all studies, instead of relying on the perspec-

tive of a single individual or role, multiple practitioners having different roles in the company were interviewed and involved in the workshops, to avoid any bias. Thus, through triangulation using multiple practitioners from different teams the threat of having a biased perspective of the processes and challenges in the organization was reduced.

Construct validity

The threats to construct validity reflect whether the measures used really represent the intended purpose of investigation [63].

For case studies, a conscious attempt was made to strive to have data source triangulation. For example, interviews and workshop results were compared with archival data, process documentation and source artifact analysis to strengthen the evidence generated in these studies. Using the appropriate amount of raw data and through a clear chain of evidence (maintaining traceability between the results, data and data sources), this validity threat was minimized.

For the systematic review reported in Chapter 2, rigor and relevance were evaluated based on what has been reported in the articles, hence few studies could potentially score higher, especially those based on Ph.D. theses. That is, the authors could have followed the steps, but due to page restrictions did not report on the results. Furthermore, an absolute scale was used in the evaluation of rigor of studies and validation of the models without compensating for the purpose of the studies. However, the principle conclusion of the chapter would not be different.

Regarding the guidelines to conduct an SPSM based study presented in Chapter 4, given that both researchers had no previous experience of SPSM, they ideally reflected the situation of a practitioner who is faced with the task of conducting an SPSM study. This lack of prior knowledge increased the likelihood that the successful outcome of this study can be attributed to the consolidated process, and not the expertise authors had in the area. However, there could be various confounding factors that the authors could not control in this study. For example, there is a threat of maturation that the authors acquired more knowledge from literature beyond what is documented in the reported guidelines.

Similarly, a majority of the process guidelines consolidated in Chapter 4 have been developed and used for system dynamics based SPSM studies. Therefore, there is a high likelihood that the consolidated process is biased in support for use of this approach. Furthermore, because of various reasons (as discussed in Chapter 4) system dynamics was the approach of choice. This means that the usefulness of the consolidated process needs to be evaluated for other simulation approaches.

For studies reported in Chapters 5 and 6, it was clearly shown that the improvements led to the identification of new improvement opportunities, practitioners developed new insights about their development processes and that the improvements resulting from the intervention will improve the quality of the software they develop and are realistic to implement. This can be attributed to the use of the proposed intervention as practitioners expressed this opinion in an anonymous survey. This confidence in the results was further strengthened in a follow-up in the organization six months after the conclusion of the study, since all the improvements have either been adopted or they are in the process of implementing them. However, there is still a need to revisit the case organization again and assess if the improvements yielded a quantifiable improvement in achieving the stated goal that was to reduce the time-to-market significantly.

External validity

The threats to external validity limit the generalization of the findings of the study outside the studied case [63].

Given that case study research has been the primary method of investigation, the results are strongly bound to the context of the studies. However, each chapter attempts to report the context of the study in detail to support generalization to this specific context. Researchers and practitioners working in a similar context may find the results transferable to their unique context [51]. In general, the results of this thesis will be interesting for organizations developing large-scale software development and trying to adapt and scale Agile software development for their context.

In Chapter 7, the strategies for study selection have been evaluated in only one systematic review. Often the title and abstract of an article is insufficient to draw a conclusion about relevance of a study e.g. the context whether a study was done in practice or in a lab is unclear. Hence, the guidelines to approach selection and how to document the decisions are applicable in other reviews regardless of the topic.

1.4 Overview of chapters

The contribution of individual chapters towards the aim of the thesis was discussed in Section 1.3.1 and illustrated in Figure 1.4. In the following sections, a summary of motivation, research method, and main results for each of the chapters is presented.

1.4.1 Chapter 2: A systematic literature review on the industrial use of software process simulation

To assess the use of simulation to support VSM in the SE context, we first explored literature on the use of simulation in the broader context of software process simulation. We found conflicting claims about the practical usefulness of SPSM, ranging from: "SPSM is useful in SE practice and has had an industrial impact" [85], "SPSM is useful however it is yet to have a significant industrial impact" [36, 25, 9] to "questions about not only the usefulness but also the likelihood and potential of being useful for the software industry" [52].

There is considerable literature on SPSM and there are a number of mapping studies [87] [86], but none is aggregating evidence. Thus, the need to evaluate evidence to assess these conflicting standpoints on the usefulness of SPSM was identified. A systematic literature review was performed to identify, assess and aggregate empirical evidence on the usefulness of SPSM. To assess the strength of evidence regarding the usefulness of SPSM for the proposed purpose, the articles were assessed based on scientific rigor and industrial relevance [26], moreover, the simulation model's credibility (in terms of the level of verification and validation (V&V) performed, see Chapter 4 for details of V&V in the context of SPSM) was taken into account.

The results of the review revealed that to date, the persistent trend is that of proof-of-concept applications of software process simulation for various purposes (e.g. estimation, training, process improvement, etc.) using a variety of simulation approaches (most commonly system dynamics and discrete event simulation). The scope of the simulation models is usually restricted to a single phase of the life-cycle or the life-cycle of a single product. A broader scope encompassing long-term product evolution, multiple product releases or complex concurrent development scenarios are rarely investigated in real-world SPSM studies.

The 87 shortlisted primary studies, scored poorly on the stated quality criteria. Also, only a few studies report some initial evaluation of the simulation models for the intended purposes. Only 18% of the primary studies verified both the model structure (i.e. representativeness of the simulation model of the software process being studied) and behavior (i.e. whether model behavior accurately captures dynamic process behavior). Overall, 13% provided an evaluation. Only 5% of the studies scored high on either rigor or relevance. Furthermore, the evaluation done was at best only static according to the definition from Gorschek et al. [20], i.e. feedback from practitioners was collected whether the model has the potential to fulfill the intended purpose. Of the overall set, only one article reports having verified structure and behavior of the model, and evaluated it against the specified purpose.

The results clearly show a lack of conclusive evidence to substantiate the claimed usefulness of SPSM for any of the intended purposes. A few studies that report the cost of applying simulation do not support the claim that it is an inexpensive method. There is no evidence to support the claims of industry adoption of SPSM and its impact on industrial practice [84, 85]. There are no reported cases of the transfer of technology where SPSM was successfully transferred to practitioners in the software industry. Furthermore, there are no studies reporting long-term use of SPSM in practice.

Key findings: The claims about the usefulness of SPSM, its industrial impact and it being an inexpensive method could not be substantiated from over three decades of research. While the reporting quality of research, and model validation steps need considerable improvement, the need for evaluating SPSM with respect to its purpose is paramount. In use of SPSM as a scientific method of inquiry, care should be taken when interpreting the results taking into account the limitations of simulation in general and the SE context in particular.

1.4.2 Chapter 3: Use and evaluation of simulation for software process education: a case study

The findings in Chapter 2 indicate that while the usefulness of SPSM for industry could not be established there are enough successful proof-of-concept applications that indicate its potential. Furthermore, even the skeptics [52] consider that SPSM has the most potential as a training and learning tool. This motivated us to evaluate the use of SPSM based intervention for an educational purpose.

Software Engineering being an applied discipline, its concepts are difficult to grasp solely at a theoretical level. In the context of a project management course, which aims to convey to students with hands-on experience the knowledge to prepare, execute and finalize a software project, we had observed that students encounter difficulties in choosing and justifying an appropriate software development process. We hypothesized that the students had inadequate experience of different software development processes, and therefore lacked the analytical insight required to choose a process appropriate for the characteristics of the course project.

This chapter presents the evaluation of the use of a simulation based game called SimSE [45] for improving students' understanding of software development processes. The effects of the intervention were measured with the Evidence-Based Reasoning framework [13] by evaluating the strength of students' arguments for choosing a particular development process over others.

The framework enabled decomposition of arguments into distinct parts, which ensured an objective evaluation of the strength of the arguments in student reports. This

assessment allowed us to gauge students' understanding of software development processes.

The results indicate that students generally have difficulty providing strong arguments for their choice of process models. Nevertheless, the assessment indicates that the intervention of the SPSM based game had a positive impact on the students' arguments. The indications reported in this chapter (from use of software process simulation in an active course) add to the confidence in evidence reported in earlier empirical studies in controlled settings. Given the potential gains as seen in this study, and the relative maturity, intuitive user interface and documentation of the SimSE tool (which was used in the study) [45], the minor additional cost of including it in a course to reinforce concepts already learned was well justified.

Key findings: Evidence from an existing systematic review and the case study presented here shows that simulation could successfully be used in combination with other teaching methods. Simulation was successfully used to reinforce already acquired knowledge about software development process models. The results indicate that while industry uptake of SPSM may be a distant dream there is evidence to encourage its use in SE education.

1.4.3 Chapter 4: Aggregating software process simulation guidelines: Literature review and action research

To combine SPSM with VSM and use it in industry, it was necessary to understand how SPSM based studies are conducted in an applied setting. However, in literature on SPSM, a number of processes were found that were positioned to target a certain simulation approach, size of the organization, modelers' experience etc. Therefore, it was not as obvious to know which of these processes to follow. This problem was addressed by conducting a literature review to identify existing process prescriptions, and analyzing and consolidating them. It was found that a common trait among all the prescribed process was the iterative and incremental nature of the process to conduct a simulation based study. The consolidation of these processes led to a six-step process. Furthermore, this consolidated process was supplemented with guidelines from SPSM literature in particular and simulation literature in general. Chapter 4 presents a brief description of each step, the sources which recommend them, guidelines applicable for each of the steps and references to relevant literature.

This process was successfully used to develop a system dynamics based simulation model of the test process at the case company. The experience and reflection on using the consolidated process for developing the simulation model are also reported in this chapter. The simulation based study was preceded by a case study at the com-

pany (using process documentation, defect database and interviews with practitioners having various roles in the organization as data sources), which provided the necessary understanding of the organization's development context and in particular their testing process.

Key findings: The overall process for conducting an SPSM based study in industry is independent of the simulation approach, experience level of modelers, and the size of the organization where the study is conducted. Another key result is the recognition that *SPSM is simulation too* i.e.while the SPSM community has redone some of the work already accomplished by other disciplines (which have a relatively longer tradition of using simulation as a method of inquiry), there is an opportunity to learn from them and leverage the similarities and address the peculiarities of the SE context.

1.4.4 Chapter 5: Evaluation of simulation assisted value stream mapping: two industrial cases

VSM as a tool for lean development has led to significant improvements in different industries. In a few studies, it has been successfully applied in a software engineering context. However, some shortcomings have been observed in particular failing to capture the dynamic nature of the software process to evaluate improvements i.e. such improvements and target values are based on idealistic situations. The aim is to overcome these shortcomings of VSM by combining it with SPSM, and to provide reflections on the process of conducting VSM with SPSM.

Chapters 2 and 3 provide an awareness of the possible ways that SPSM can be utilized, what techniques and approaches exist, and which ones are more appropriate for a certain purpose and for the scope of modeling. The evidence gathered in these chapters provides confidence and bounds the expectations regarding the utility of SPSM. Likewise, Chapter 4 provides the guidelines to facilitate the use of SPSM in industry. Thus, the work presented in Chapters 2 to 4 provides the foundation, and together it facilitates conscious decision making about choosing an appropriate modeling purpose, scope, and the steps to reach the required level of model credibility.

In this chapter, results of a literature review on how other disciples have combined VSM with simulation are presented. These are used to propose how SPSM can be combined with VSM using the guidelines adapted for the SE context by Khurum et al. [31]. Using case study research, VSM assisted by SPSM was used for two products at Ericsson AB, Sweden. Ten workshops were conducted in this regard. Simulation in this study was used as a tool to support discussions instead of as a prediction tool. The results have been evaluated from the perspective of the participating practitioners, an

external observer, and reflections of the researchers conducting the simulation that was elicited by the external observer.

Significant constraints hindering the product development from reaching the stated improvement goals for shorter lead-time were identified. The use of simulation was particularly helpful in having more insightful discussions and to challenge assumptions about the likely impact of improvements. However, simulation results alone were found insufficient to emphasize the importance of reducing waiting times and variations in the process. Due to the participatory approach followed in VSM, with involvement of various stakeholders, consensus building steps, emphasis on flow (through waiting time and variance analysis) and the use of simulation led to *realistic* improvements with a high *likelihood* of implementation.

Key findings: Use of simulation proved an effective tool to prompt insightful discussions and reflect on the impact of addressing certain challenges and for reflecting on the likely benefits of proposed improvements. The results clearly indicate positive experience from using simulation to assist VSM, and practitioners would like to see more extensive use of simulation as a decision support tool. However, researchers had to perform the simulation model building activities and present the models and their outputs, and it is difficult to foresee practitioners developing simulation models on their own in the future, at least in the case company. Another major limitation to the utility of simulation is a lack of data, simulation results did not carry the same weight even when calibrated using within-company data from a different team.

1.4.5 Chapter 6: FLOW-assisted value stream mapping in a large-scale software product development

The current adaptations of VSM from Lean manufacturing focus mostly on the flow of artifacts and have taken no account of the important information flow in software development. A solution specifically targeted towards information flow elicitation and modeling is FLOW. This chapter proposes and evaluates the combination of VSM and FLOW to identify and alleviate information and communication related challenges in large-scale software development.

Using case study research, FLOW assisted VSM was used for a large product at Ericsson AB, Sweden. Both the process and the outcome of FLOW assisted VSM have been evaluated from the practitioners' perspective. It was found that FLOW helped to systematically identify and elaborate waste, challenges and improvements related to information flow. Practitioners were positive about the use of VSM as a method of choice for process improvement and found the overview of the entire process using the FLOW notation very useful. The combination of FLOW and VSM presented in this study was

successful in uncovering issues and systematically characterizing their solutions, indicating their practical usefulness for waste removal with a focus on information flow related issues.

Key findings: FLOW provides a systematic approach to elicit, visualize and analyze information flows. The lightweight approach with intuitive notation provided the necessary support missing in the current VSM to analyze information flow and effectively streamline the value stream.

1.4.6 Chapter 7: Identifying and evaluating strategies for study selection in systematic literature studies

The study selection is critical to improve the reliability of secondary studies. However, when designing the review protocol for the systematic review presented in Chaper 2 of this thesis, a lack of guidelines to systematically perform and document study selection was identified. By analyzing the existing systematic literature reviews, the strategies and rules employed by researchers for ensuring reliability of selection were identified.

Thirteen different strategies for inclusion and exclusion were identified. Three were used to assure objective inclusion/exclusion criteria to reduce bias, three to resolve disagreements and uncertainties due to bias, and seven defined decision rules on how to act based on disagreements/agreements. It was also found that the strategies most frequently used in existing reviews were the ones proposed in the systematic review guidelines [34].

To evaluate the identified strategies, a process for study selection employing them was presented and evaluated in an SLR (reported in Chapter 2 on the usefulness of software process simulation). As expected, it was found that the most effective strategy is the most inclusive strategy, as all less inclusive strategies lead to a loss of relevant papers. From an efficiency point of view adaptive reading (proposed in this study) allows following the most inclusive strategy with relatively little additional effort. The importance of having multiple reviewers and to interpret inter-rater agreement statistics carefully was also highlighted quantitatively. Other interesting observations were related to the impact of following less inclusive strategies on the results of the study. However, no conclusions can be drawn in that regard.

Key findings: Importance of having more than one reviewer to improve reliability of secondary studies. Improve repeatability by clearly documenting the selection criteria, decisions and the process, and pilot and use inter-rater statistics to assess the quality. However, results of pilots and inter-rater statistics should be interpreted with care, given the high likelihood of noise in database-search for relevant articles.

1.5 Discussion

1.5.1 VSM with simulation

We evaluated the use of SPSM to support VSM in two industrial cases. Inspired by the ways SPSM has been used to support VSM in other disciplines, a framework listing the possible uses along with references to relevant literature were provided. Furthermore, the use of simulation to support discussions regarding which challenges should be addressed on priority and under the given assumptions what is the impact of proposed improvements. SPSM contributed substantially in highlighting the potential of reducing waiting times and improving the flow in the process. The ability to model uncertainty in the process was visibly the most beneficial in providing a realistic test bed to assess the improvements by making the assumptions of practitioners explicit and providing input for choosing between alternatives.

However, there are several considerations when utilizing SPSM, which are discussed in the following subsections.

Cost of SPSM

Compared to experimenting with the actual process SPSM is positioned as an inexpensive mechanism of inquiry [81] [42] [30] [38]. However, the premise that a simulation model produces as reliable results as a case study, where the change is introduced on a smaller scale, has two problems:

1. The cost of conducting an effective simulation should be considered. It will include, e.g.:
    - The cost of the necessary measurement program that can feed the model with accurate data (as seen in Chapter 5).
    - The cost in terms of required tool support, training and effort for development and maintenance of simulation models.
2. When comparing to other methods of studying a system (like experiment or case study research) the strength of evidence should also be taken into account.

In the SLR of SPSM literature (see Chapter 2), only two studies reported an estimate of the effort spent on the simulation based study. Pfahl and Lebsanft [55] report 18 months of calendar time for the simulation study and an effort of one person year in consulting and 0.25 person years for the development part. In another study, Pfahl [53] only reports the calendar time of three months for knowledge elicitation and modeling

and four meetings with the client. Shannon [65] predicted a high cost for the simulation as well, *"a practitioner is required to have about 720 hours of formal classroom instruction plus 1440 hours of outside study (more than one man-year of effort)"*.

Given the nature of software development where change is so frequent (technology changes, software development process, environment, and customer requirements etc.), it is very likely that for the simulation as a decision support tool will need adaptation. Expecting practitioners to put one man-year of effort is very unrealistic. Thus, there is a need to reduce this cost, identify and alleviate other challenges that practitioners face to develop and maintain simulation models on their own. Murphy and Perera [44] have done some research in this direction in the automotive and aerospace industry where they looked at the problems encountered and enabling factors for a successful implementation of a simulation in a company. Similarly, when we using simulation as a decision support tool for use by practitioners it raises unique requirements on the simulation tools, e.g. integration of the simulation model with the existing decision support system [6] [44].

Evidence for the usefulness of simulation

Apart from the cost of using simulation another important aspect that will highly influence the decision to adopt simulation in practice is the evidence of the usefulness of SPSM for its intended purposes in the real-world software development.

Here is a brief summary of promised benefits of software process simulation (for details see Chapter 2 for applied SPSM and for overall SPSM literature see [87] and [86]):

- Improved effort and cost estimation
- Improved reliability predictions
- Improved resource allocation
- Risk assessment
- Studying success factors for global software development
- Technology evaluation and adoption
- Training and learning

Christie's [14] words of caution that "simulation is not a panacea" are not repeated often enough. And simulation is presented as exactly that: a panacea, a silver bullet. However, when an attempt was made in this thesis to assess the evidence presented in literature for these claimed benefits, it was found that there is almost no evaluation reported. Furthermore, the studies reported in SPSM literature scored poorly in terms of scientific rigor (for a detailed discussion see Chapter 2). The lack of discussion of

limitations or threats to the validity of findings from the SPSM studies (see Chapter 2) further highlights this problem.

Whereas to establish SPSM as an alternative to static process models, analytical models, and as means of planning, management and training in industry, the evidence of its efficacy must be presented. Unfortunately, based on the systematic review (see Chapter 2) and the survey results [3], SPSM modelers see verification and validation as the evaluation of simulation models. Some initial work towards a framework that highlights the need for evaluation of the value aspect of a simulation model is done by Ahmed et al. [2].

In the thesis, the usefulness of SPSM was evaluated in both industrial and academic contexts. In an industrial context, we evaluated the usefulness of simulation as a tool for reflection and reasoning about changes to the software development process. In Chapter 5, SPSM was successfully utilized to reflect on the impact of addressing identified challenges and reason about which improvements should be prioritized to implement in the context of VSM.

For software process education, based on results of a systematic literature review [74] and the work presented in Chapter 3, SPSM is considered a useful tool to reinforce the theoretical understanding of software process models. SPSM achieves this by putting the knowledge to practice in a simulated environment. It must be highlighted that SPSM has shown value as a supplement to existing teaching methods and should not be considered an alternative to existing methods. For example, in the case reported in Chapter 3, it would take an impractical number of iterations to learn a process model just by playing the SPSM based game repeatedly. This would not be an efficient approach to learning.

Moreover, experiencing process models in a simulated environment does not mean that we can eliminate the need for the practical project (where the students choose and apply a process model to develop a software product). SPSM based games impose certain restrictions on how the games are played. Furthermore, the underlying process models implemented in such games have a fixed interpretation with assumptions about how various factors are interrelated. For example, in the simulation tool used in Chapter 3, students will not have the freedom to combine best practices from process model "A" with another process model say "B", which may otherwise make "B" more appropriate for the given context. Of course, such a combination can be modeled but that would mean training the students in simulation modeling which is beyond the scope of the course.

Strength of evidence generated by SPSM

Another aspect is strength of evidence generated by SPSM. This aspect is indirectly evaluated in the evidence of usefulness but in this section the focus is on more intrinsic elements that are the foundation of SPSM. A critical discussion of the limitations of SPSM will help position it appropriately as a means to supplement other empirical methods. Some limitations discussed below are generally true for any simulation but they are aggravated in SPSM because of the nature of software development.

Models are simplification: Simulation models like any models are a simplification, abstraction, and at best approximations of the system of interest [14]. In large organizations, the documented process, the actual process and the perceived process are different. Thus, there is some level of uncertainty brought into the software process simulation modeling. Looking at the existing literature a lot of the studies simulate process models based on standards (e.g. IEEE-12207) claiming that this is the process used by the company. The processes described in standards are often adapted in practice, and not applied as specified in a standard, which is a potential validity threat to the conclusions drawn from such studies. Similarly, this raises questions about the validity of the simulation results when the models are only calibrated with data from the case company using an existing process model.

In Chapter 4, for example, when we were simulating the test process for training purposes, we could not just use a general test process like the ISO/IEC/IEEE 29119-2. It was important to capture the testing process followed in the company in sufficient detail. So, that the practitioners can relate with the simulation model, trust its results, and find the lessons learned from working with the simulation model transferable to practice. This could only be achieved if the model sufficiently replicates both the structure (e.g. tasks, activities and their sequence) and behavior (e.g. data about defect density, number and frequency of iteration in various phases) of the real process. However, besides all the efforts to make the model as realistic as possible, due to the number of assumptions and reliance on "estimation" for the calibration of the model for missing data, the model can only be used for illustrative purposes.

Lack of evidence for causal relations: There are no established physical laws that govern software development and SPSM deals with entities that are difficult to quantify [14]. Confidence in the model depends on verification and validation of the structure and behavior of the model [14]. Similarly, our existing understanding of the causal relations is not at the level of physical laws, e.g. on size and effort [27], and for software metrics validation in general [32]. This brings another layer of uncertainty in the software process simulation models.

In Chapter 5, to support the simulation modeling we tried to identify relationships between several challenges identified in the process and the factors influencing them.

However, it was an unsuccessful attempt, as soon it became a speculative exercise where practitioners reasoned about how several factors were inter-related. Finally, the practitioners were not confident at all in quantifying (even as estimates) these relations.

Lack of data: Olsen [46] describes that a model without supporting data is a little more than a visual metaphor. He further highlights the importance of accurate measures to enable adequate predictions by the model [46]. However, a lack of empirical data in the software industry is a common challenge faced by software process simulation modelers [30]. An often used solution in the existing SPSM research is the use of industry averages and estimates from analytical models. Given the context driven nature of software development [51] it is clear that this adds a certain amount of uncertainty in the software process simulation models that are calibrated with cross-company data. Another means often used to overcome the lack of data is to use the analytical models like COCOMO [12]. Ironically, the analytical models that were criticized for their static nature [30] are being used for calibrating the simulation models. Furthermore, there is inconclusive evidence for the usefulness of cross company data, e.g. Kitchenham et al. [35] found evidence both supporting and refuting the usefulness of cross-company data sets for cost estimation. Given that a simulation cannot be effective unless both the model and data accurately represent the real-world [66]. The lack of data, therefore, adds another layer of uncertainty in software process simulations.

In Chapters 4 and 5, it was realized that SPSM making opportunistic use of available data (i.e. where the measurement program collecting data in an organization was not directed by the needs of the simulation model) added serious shortcomings to the calibration of the models. For example, in the study reported in Chapter 4 while the detailed data for attributes of requirements being implemented in each version of the product and the defects found in each version were available, there was no traceability between requirements and defects. Thus, neither the relation between e.g. the size of the requirements and number of defects introduced nor the number of times a requirement had to iterate in various phases before being released could be discerned from the data. Therefore, we had to rely on expert judgment to estimate and calibrate the model, but the variation between various experts' opinion added serious threats to the validity of the reliability of the results beyond an illustrative tool for training.

Similarly, in the case studies reported in Chapter 5, simulation was used as a tool for reflection and reasoning about changes in the development process. It was clear that simulation results led to more insightful discussions in both cases. However, in one product (due to the unavailability of data) data from the other product were used for the calibration of the model. In this case, practitioners argued that the results were less applicable for their case.

In other cases where practitioners generally have a dismissive attitude of "not applicable here" towards empirical evidence [59], it is likely that such practitioners will

dismiss the results of simulation altogether if it were not calibrated with their data. The likelihood of such reactions is even higher if the simulation results do not align with their perception. Thus, a lack of data threatens the usefulness of SPSM even as a tool for reflection.

Simulation as an alternative empirical method

Simulation is motivated as an alternative by citing the high cost and risk of manipulating the actual system [42] [81]. In principle this statement is true, but that is precisely the reason why pilot studies are performed where the change is incorporated on a small scale.

Simulation is sometimes suggested as an alternative to case study as the results of a simulation study are somehow more generalizable. For example, *"the usual way to analyze process behaviour is to perform the actual process in a case study and observe the results. This is a very costly way to perform process analysis, because it involves the active participation of engineers. Furthermore, results from a particular case study cannot necessarily be generalized to other contexts. Another way of analyzing processes is to simulate them"* [66]. Others have proposed it as an alternative to expensive controlled experiments that are often done at a scaled down problem in laboratory settings. For example, *"Simulation modeling provides a viable experimentation tool for such a task. In addition to permitting less costly and less time-consuming experimentation, simulation-type models make "perfectly" controlled experimentation possible"* [1].

These observations indicate that the SPSM researchers view simulation as a research method. However, there is an evident desire to transfer the SPSM state-of-the-art to the software industry considering the SPSM studies reported in literature that involved companies.

Based on the results of a systematic literature review where the claims of usefulness of SPSM could not be substantiated (see Chapter 2), our own experience of conducting SPSM based studies in industrial settings (see Chapters 4 and 5) and the weaknesses of SPSM as discussed in Section 1.5.1 we do not consider simulation models an alternative to other research methods.

Likelihood of widespread adoption

The cost of simulation (as discussed in Section 1.5.1) and the strength of evidence (see Section 1.5.1) generated by SPSM should inform our decision of when and how to use SPSM. Based on the research presented in this thesis, characterizing the limitations of the utility of SPSM, we consider it a decision-support tool used by researchers and

consultants to support practitioners. No reported cases of long-term use of SPSM or adoption by industry have been found (see Chapter 3). Also, our own experience of using simulation in industry (see Chapter 5) indicates very little likelihood of wide spread adoption of SPSM in the software industry. For example, practitioners participating in the studies reported in Chaper 5 acknowledged the contribution of simulation in the VSM related work, and expressed the desire to see it being more extensively used for decision support. However, they were not willing to model themselves and did not allocate time to receive trainings even though we offered to help.

Conducting SPSM based study

The analysis of existing process guidelines for conducting SPSM revealed considerable similarity among them. It was further concluded that the process for conducting an SPSM based study is independent of the factors that were highlighted as the focus of each of these processes e.g. simulation approach, modelers experience or organizational size. Furthermore, at an abstract level the process is very similar to the generally recommended process for simulation based studies in other disciplines. This suggests that SPSM and simulation in other disciplines are more similar than previously thought. We can learn from their methodology, reporting guidelines, best practices and lessons learned and evaluate them for SPSM.

The consolidated process for conducting an SPSM based study in industry presented in this thesis (see Chapter 4) reduces confusion in choice of a process caused by the plethora of process prescriptions. The process can act as a checklist for advanced simulation modelers as well as facilitate novice modelers for conducting SPSM based studies. The utility of this model was demonstrated successfully in this thesis.

1.5.2 VSM with FLOW

Software development is a knowledge intensive activity [10]. However, process models for information flow often neglect verbal, informal communication or the experience required to carry out complex software development tasks. Instead, such models mainly focus on only documented information flow [60]. In practice, for example, many requirements, decisions, and rationales are only communicated using informal means. Thus, it is important to capture both documented (“solid”) and undocumented, verbal, or informal (“fluid”) flow of information [69] [64].

Similarly, the current push towards these more flexible approaches further increases the importance of capturing the “fluid” information flow. Since, traditional plan-driven approaches tend to rely on documented information propagation, while the Agile approaches rely on “direct communication” [15].

Solid information is typically stored in documents, files, recordings, or other kinds of permanent media. Creating and retrieving information takes longer and requires more effort, but does not rely on the creator, and can be performed at a later point in time and by many people. Fluid information flows faster and often more easily: phone calls, chats, or short messages do not get formally documented, but convey information as well. Fluid information is typically stored in a person's mind. It is important to see both solid and fluid information as inseparable and mutually constituting [24]. Each has its own strengths and weaknesses but their interplay complements each other. Many companies need to mix and adjust elements from both ends of the spectrum and try to find an overall optimum. FLOW is particularly useful in this regard as it showed that it can help organizations take those decisions on a case by case basis.

Both VSM and FLOW have synergies and hence are even more successful in combination. FLOW was successfully used to elicit and model information flows. The approach and the analysis in the VSM. While VSM provided a broader framework that brings a systematic approach of reflecting on the value stream to eliminate waste and identifying improvements. Furthermore, the artifact flow analysis, where the time-line for the value stream is assessed, helps to quantify the implications of challenges in information flow.

1.6 Some indications of research impact

Apart from three cases of conducting VSM reported in this thesis (Chapters 5 and 6), in the fourth instance, a practitioner from the case company (who was involved in the earlier VSM studies) took over the entire responsibility for conducting VSM. We were involved only in a supportive role (doing the data analysis in the background). Since, then the practitioner has been involved in training other employees in sites in Germany, China, India and the US to facilitate and encourage local teams to start reporting the new measurements, use the dashboards and conduct VSM locally.

So, far they have conducted VSM for eight products and the feedback has been very positive from the teams. They consider that the benefits of the workshops outweigh the time spent by participating in them. Even with product teams with skeptical participants the value became apparent once they had participated in the workshops.

All the reports of quantitative data analysis on lead-time, and the visualizations for flow (used during the work that is reported in Chapters 5 and 6) are now part of the VSM dashboard to facilitate the practitioners to analyze their development processes. There are new measurement points that have been introduced into the process based on the work presented in the thesis.

Feedback on dashboard-based reports is also positive as practitioners found it useful to reflect objectively on the perceptions that they had about the state of their product development (e.g. size and place of backlogs). However, the dashboard is not currently used as part of routine work and they are hoping that having local *"VSM facilitators"* will help to get VSM rolling in the organization.

The feedback on incorporating simulation has also been positive, but there is a need to make it more accessible to practitioners that will take on the role of *"VSM facilitators"* in the organization. Given the consistent overall process that is followed throughout the organization and the standard suite of measurements that are collected in each of the products provides some avenues for the use of simulation. The expectation is to explore further the possibility to train the *"VSM facilitators"* to enable the use of simulation models even though they will not be required to create the models themselves. This would provide them with the tool set to perform certain analyses and highlight the importance of having continuous/even flow over the typical resource utilization mindset that is currently prevalent.

Similarly, there has been some influence of the work presented in Chapter 3 within our department at the university. The simulation based intervention used in that study is part of the graduate level course as a four-hour workshop distributed over two days. Furthermore, some colleagues have used the Evidence-Based Reasoning framework [13] identified and adapted in Chapter 3 to give students formative feedback in scientific argumentation.

The work presented in Chapter 7 to improve the guidelines for systematic secondary studies has been well received by the community, it was published as two short papers. The first paper [48] received high scores from authors of the guidelines for SLR in SE and is part of the updated guidelines [33] the evaluation paper [4] got the best short paper award at the International Symposium on Empirical Software Engineering and Measurement, Turin, Italy 2014.

1.7 Conclusion

This thesis aimed to facilitate the adoption of Lean thinking in the software industry. We evaluated VSM to operationalize Lean thinking using SPSM and FLOW. To incorporate SPSM into VSM, there was a need: to evaluate the usefulness of SPSM in the SE context, to develop guidelines for conducting simulation based study in industry, and provide a framework to combine SPSM with VSM. Similarly, a proposal to combine FLOW with VSM to facilitate information flow analysis was presented and evaluated. Thus, both artifact flow (or material flow) and information flow analysis in VSM were improved. Based on the findings of this thesis, the following conclusions are drawn:

RQ: How to operationalize Lean thinking in the context of software development?

In current adaptations of Lean in SE, VSM is either a tool that is used only after a certain set of practices in a prescribed order have been implemented [22] or its role is limited to visualizing the process and eliminating waste [58]. This may explain why besides the popularity and interest in Lean, only a few applications of this key practice have been reported in the SE context [47] [73]. In this thesis, value stream mapping was identified as the key practice that can operationalize Lean thinking and facilitate the organizations' attempting a Lean transformation. Contrary to some prescribed order for adopting Lean practices [22], which is highly unlikely to be applicable/relevant in every context, VSM provides a systematic approach to identify the most relevant changes in each unique context. Furthermore, while VSM directly targets waste and by identifying and eliminating it, it directly contributes to achieving the aims of pull, flow, continuous improvement and above all to "optimize the whole".

The Lean principles are used to guide and help practitioners applying VSM to identify waste and recommend improvements, where the unique characteristics of development teams, software product, and other technical and organizational factors are taken into consideration. Similarly, there are no fixed definitions of waste, the categories identified in literature [57] are an excellent starting point and practitioners may use them for reflecting on their own development process when doing VSM. The guidelines that were further improved in this thesis, take some key lessons learned from software process improvement literature into consideration. With the use of VSM we were able to get the buy-in from both management and development teams. This was evident from their confidence in the value of the improvements in terms of improving the quality of the product and that improvements are realistic and will be implemented. It was also confirmed in a follow-up workshop that virtually all suggested improvements have been implemented.

Sub-RQ-1: How to improve artifact flow analysis in value stream mapping? The combination of simulation with VSM allowed overcoming the current limitation of VSM using static models for identification of challenges and selection of improvement alternatives to implement. In the given case, simulation contributed significantly to help challenge whether addressing the perceived major challenges had the potential of achieving the target of improvement. Simulation output provided useful input for discussions, to make our assumptions explicit and helped to make a more informed decision about the choice of improvement. Overall, by taking the uncertainty of the process into the model we achieved a quantitative data source for triangulation of expert opinion beyond the empirical data about the current process performance. Possible ways of combining simulation with VSM and their likely benefits have been identified

in the thesis. However, a decision has to be made on a case-to-case about the feasibility of using simulation.

Sub-RQ-2: How to improve information flow analysis in value stream mapping? There are a number of information flow modeling alternatives that are available, however, FLOW was found to be a light-weight and systematic approach to elicit information and visualize information flows using its simple and intuitive notation. In the context of VSM, it enabled to capture the process in sufficient detail to expose challenges related to information flow e.g. overload on key stakeholders, and sole reliance on a human resource as an information source while multiple non-collocated sites require the information. In our proposal, FLOW works in the background without the need to provide practitioners any additional training or familiarization with the methodology of FLOW. This also made the combination with VSM a success.

1.8 Future work

There is considerable interest in scaling Agile for large-scale software development. We intend to continue doing research on the topic. Application of Lean thinking to software development is relatively new however, there is considerable research that is complementary to generic lean principles. We intend to further leverage from existing work in the complementary areas of value-based software engineering [11], theory of constraints [5], and systems thinking and system dynamics [37] [17].

Seeing the benefits of using simulation, there is an interest from our industrial partner for exploring further avenues for its use. The future direction of our research with regard to simulation is to identify opportunities for the use of SPSM where practitioners can benefit from its use without having to invest extensively in learning to build simulation models. One such opportunity exists in the case company, where a shared process and measurements are implemented that provides a consistent view of the current development activities.

Similarly, we will continue to contribute to the development of empirical methods for SE research. One proposal is to assess and improve the guidelines for doing database search or snowball sampling employing graph theory.

References

[1] T. K. Abdel-Hamid, "Understanding the "90% syndrome" in software project management: A simulation-based case study," *Journal of Systems and Software*, vol. 8, no. 4, pp. 319–330, 1988.

[2] R. Ahmed, T. Hall, and P. Wernick, "A proposed framework for evaluating software process simulation models," in *Proceedings of the International Workshop on Software Process Simulation and Modeling (ProSim)*, 2003.

[3] R. Ahmed, T. Hall, P. Wernick, S. Robinson, and M. Shah, "Software process simulation modelling: A survey of practice," *Journal of Simulation*, vol. 2, no. 2, pp. 91–102, 2008.

[4] N. B. Ali and K. Petersen, "Evaluating strategies for study selection in systematic literature studies," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '14. New York, NY, USA: ACM, 2014, pp. 45:1–45:4.

[5] D. J. Anderson, *Agile management for software engineering: Applying the theory of constraints for business results*. Prentice Hall Professional, 2003.

[6] O. Balci, "Guidelines for successful simluation studies," in *Proceedings of the Winter Simulation Conference*. IEEE Press, 1990, pp. 25–32.

[7] J. Banks, "Introduction to simulation," in *Proceedings of the Winter Simulation Conference*, ser. WSC '99. New York, NY, USA: ACM, 1999, pp. 7–13.

[8] B. Berenbach and G. Borotto, "Metrics for model driven requirements development," in *Proceedings of the 28th International Conference on Software Engineering*, ser. ICSE '06. New York, NY, USA: ACM, 2006, pp. 445–451.

[9] T. Birkhölzer, "Software process simulation is simulation too - what can be learned from other domains of simulation?" in *Proceedings of the International Conference on Software and System Process (ICSSP)*. IEEE, 2012, pp. 223–225.

[10] F. O. Bjørnson and T. Dingsøyr, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used," *Information and Software Technology*, vol. 50, no. 11, pp. 1055–1068, 2008.

[11] B. W. Boehm, "Value-based software engineering: Overview and agenda," in *Value-based software engineering*. Springer, 2006, pp. 3–14.

[12] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. J. Madachy, D. J. Reifer, and B. Steece, *Software cost estimation with COCOMO II*. Prentice Hall, Aug. 2000.

[13] N. J. S. Brown, E. M. Furtak, M. Timms, S. O. Nagashima, and M. Wilson, "The evidence-based reasoning framework: Assessing scientific reasoning," *Educational Assessment*, vol. 15, no. 3-4, pp. 123–141, 2010.

[14] A. M. Christie, "Simulation: An enabling technology in software engineering," *CROSSTALK–The Journal of Defense Software Engineering*, vol. 12, no. 4, pp. 25–30, 1999.

[15] A. Cockburn, *Agile software development*. Addison Wesley, 2002.

[16] K. R. Felizardo, G. F. Andery, F. V. Paulovich, R. Minghim, and J. C. Maldonado, "A visual analysis approach to validate the selection review of primary studies in systematic reviews," *Information and Software Technology*, vol. 54, no. 10, pp. 1079–1091, 2012.

[17] J. W. Forrester, "System dynamics, systems thinking, and soft OR," *System Dynamics Review*, vol. 10, no. 2-3, pp. 245–256, 1994.

[18] F. J. J. Fowler, *Improving survey questions : design and evaluation*. Thousand Oaks, California: Sage Publications, 1995.

[19] A. Fuggetta, "Software process: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*. ACM, 2000, pp. 25–34.

[20] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice," *IEEE Software*, vol. 23, no. 6, pp. 88–95, 2006.

[21] T. Greenhalgh and R. Peacock, "Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources," *BMJ: British Medical Journal*, vol. 331, no. 7524, p. 1064, 2005.

[22] C. Hibbs, S. Jewett, M. Sullivan, C. Hibbs, S. Jewett, and M. Sullivan, *The art of lean software development: a practical and incremental approach*. O'Reilly Media, Inc., 2009.

[23] P. Hines, M. Holweg, and N. Rich, "Learning to evolve," *International Journal of Operations & Production Management*, vol. 24, no. 10, pp. 994–1011, 2004.

[24] D. Hislop, *Knowledge management in organizations: A critical introduction*. Oxford University Press, 2013.

[25] D. Houston, "Research and practice reciprocity in software process simulation," in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 219–220.

[26] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empirical Software Engineering*, vol. 16, no. 3, pp. 365–395, 2010.

[27] M. Jørgensen and B. Kitchenham, "Interpretation problems related to the use of regression models to decide on economy of scale in software development," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2494 – 2503, 2012.

[28] S. Jalali and C. Wohlin, "Systematic literature studies: Database searches vs. backward snowballing," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2012, pp. 29–38.

[29] A. Kasoju, K. Petersen, and M. V. Mäntylä, "Analyzing an automotive testing process with evidence-based software engineering," *Information and Software Technology*, vol. 55, no. 7, pp. 1237 – 1259, 2013.

[30] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling : Why ? What ? How ?" *Journal of Systems and Software*, vol. 46, 1999.

[31] M. Khurum, K. Petersen, and T. Gorschek, "Extending value stream mapping through waste definition beyond customer perspective," *Journal of Software: Evolution and Process*, vol. 26, no. 12, pp. 1074–1105, 2014.

[32] B. Kitchenham, "What's up with software metrics? - a preliminary mapping study," *Journal of Systems and Software*, vol. 83, no. 1, pp. 37 – 51, 2010.

[33] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049–2075, 2013.

[34] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 2007.

[35] B. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus within-company cost estimation studies: A systematic review," *IEEE Transactions on Software Engineering*, vol. 33, no. 5, pp. 316–329, 2007.

[36] J. Münch, "Evolving process simulators by using validated learning," in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 226–227.

[37] R. J. Madachy, *Software process dynamics*. Wiley-IEEE Press, 2008.

[38] ——, "Simulation," in *Encyclopedia of Software Engineering*. John Wiley & Sons, Inc., 2002.

[39] J. A. McCall, G. Wong, and A. Stone, "A simulation modeling approach to understanding the software development process." in *Proceedings of the 4th Annual Software Engineering Workshop, Goddard Space Flight Center Greenbelt, Maryland*, 1979.

[40] J. McKay and P. Marshall, "The dual imperatives of action research," *Information Technology & People*, vol. 14, no. 1, pp. 46–59, 2001.

[41] H. L. McManus, *Product Development Value Stream Mapping (PDVSM) manual*, 1st ed., Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 2005.

[42] M. Melis, I. Turnu, A. Cau, and G. Concas, "Evaluating the impact of test-first programming and pair programming through software process simulation," *Software Process: Improvement and Practice*, vol. 11, no. 4, pp. 345–360, 2006.

[43] S. Mujtaba, R. Feldt, and K. Petersen, "Waste and lead time reduction in a software product customization process with value stream maps," in *Proceedings of the 21st Australian Software Engineering Conference (ASWEC)*, 2010, pp. 139–148.

[44] S. Murphy and T. Perera, "Key enablers in the development of simulation," in *Proceedings of the Winter Simulation Conference*, 2001, pp. 1429–1437.

[45] E. O. Navarro and A. Van Der Hoek, "Software process modeling for an educational software engineering simulation game," *Software Process: Improvement and Practice*, vol. 10, no. 3, pp. 311–1670, 2005.

[46] N. C. Olsen, "The software rush hour (software engineering)," *IEEE Software*, vol. 10, no. 5, pp. 29–37, 1993.

[47] J. Pernstål, R. Feldt, and T. Gorschek, "The lean gap: A review of lean approaches to large-scale software systems development," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2797–2821, 2013.

[48] K. Petersen and N. B. Ali, "Identifying strategies for study selection in systematic reviews and maps," in *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011*, 2011, pp. 351–354.

[49] K. Petersen, C. Gencel, N. Asghari, D. Baca, and S. Betz, "Action research as a model for industry-academia collaboration in the software engineering context," in *Proceedings of the 2014 International Workshop on Long-term Industrial Collaboration on Software Engineering*. ACM, 2014, pp. 55–62.

[50] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, no. 0, pp. –, 2015.

[51] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2009, pp. 401–404.

[52] D. Pfahl, "Process simulation: A tool for software project managers?" in *Software Project Management in a Changing World*, G. Ruhe and C. Wohlin, Eds. Springer Berlin Heidelberg, 2014, pp. 425–446.

[53] ——, "PL-SIM: a generic simulation model for studying strategic SPI in the automotive industry," in *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*. Citeseer, 2004.

[54] ——, *An integrated approach to simulation based learning in support of strategic and project management in software organisations*. Ph.D. dissertation, Fraunhofer-IRB-Verlag, 2001.

[55] D. Pfahl and K. Lebsanft, "Knowledge acquisition and process guidance for building system dynamics simulation models: an experience report from software industry," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 04, pp. 487–510, 2000.

[56] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineering*, vol. 13, no. 3, pp. 303–337, 2008.

[57] M. Poppendieck and T. Poppendieck, *Implementing lean software development: From concept to cash*. Addison-Wesley, 2006.

[58] ——, *Lean software development: an agile toolkit*. Addison-Wesley Professional, 2003.

[59] A. Rainer, D. Jagielska, and T. Hall, "Software engineering practice versus evidence-based software engineering research," *ACM SIGSOFT Software Engineering ...*, pp. 1–5, 2005.

[60] A. Rausch, C. Bartelt, T. Ternit, and M. Kuhrmann, "The V-Modell XT Applied - Model-Driven and Document-Centric Development," in *Proceedings of the 3rd World Congress for Software Quality*, 2005.

[61] H. Robinson, J. Segal, and H. Sharp, "Ethnographically-informed empirical studies of software practice," *Information and Software Technology*, vol. 49, no. 6, pp. 540 – 551, 2007.

[62] M. Rother and J. Shook, *Learning to see: Value stream mapping to add value and eliminate MUDA*. Brookline, Mass.: Lean Enterprise Institute, 1999.

[63] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[64] K. Schneider, K. Stapel, and E. Knauss, "Beyond documents: visualizing informal communication," in *Proceedings of the 3rd International Workshop on Requirements Engineering Visualization (REV '08)*, Barcelona, Spain, Nov. 2008.

[65] R. E. Shannon, "Intelligent simulation environments." in *Proceedings of the Conference on Intelligent Simulation Environments*, 1986, pp. 150–156.

[66] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to advanced empirical software engineering*. Springer, 2008.

[67] D. Sjøberg, A. Johnsen, and J. Solberg, "Quantifying the effect of using kanban versus scrum: A case study," *IEEE Software*, vol. 29, no. 5, pp. 47–53, Sept 2012.

[68] K. Stapel, E. Knauss, and K. Schneider, "Using FLOW to improve communication of requirements in globally distributed software projects," in *Proceedings of the International Workshop on Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*. IEEE, Aug 2009, pp. 5–14.

[69] K. Stapel and K. Schneider, "Managing knowledge on communication and information flow in global software projects," *Expert Systems*, pp. n/a–n/a, 2012.

[70] ——, "FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW," Fachgebiet Software Engineering, Leibniz Universitt Hannover, Tech. Rep., 2012.

[71] Y. Sun, Y. Yang, H. Zhang, W. Zhang, and Q. Wang, "Towards evidence-based ontology for supporting systematic literature review," in *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)*. IET, 2012, pp. 171–175.

[72] C. K. Swank, "The lean service machine," *Harvard Business Review*, vol. 81, no. 10, pp. 123–130, 2003.

[73] X. Wang, K. Conboy, and O. Cawley, ""Leagile" software development: An experience report analysis of the application of lean approaches in agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1287–1299, 2012.

[74] C. G. von Wangenheim and F. Shull, "To game or not to game?" *IEEE Software*, pp. 92–94, 2009.

[75] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quarterly*, vol. 26, no. 2, pp. xiii–xxiii, 2002.

[76] S. Winkler, "Information flow between requirement artifacts. results of an empirical study," in *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2007, pp. 232–246.

[77] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer, 2012.

[78] C. Wohlin, P. Runeson, P. A. da Mota Silveira Neto, E. Engström, I. do Carmo Machado, and E. S. de Almeida, "On the reliability of mapping studies in software engineering," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2594–2610, 2013.

[79] J. P. Womack and D. T. Jones, *Lean thinking: banish waste and create wealth in your corporation*. Simon and Schuster, 2010.

[80] J. P. Womack, D. T. Jones, and D. Roos, *Machine that changed the world*. Simon and Schuster, 1990.

[81] M. Wu and H. Yan, "Simulation in software engineering with system dynamics: A case study," *Journal of Software*, vol. 4, no. 10, pp. 1127–1135, 2009.

[82] Q. Yang, X. Chen, and T. Lu, "Case study: Simulation analysis for complex R&D project based on iteration process model," in *Proceedings of the 2011 International Conference on Instrumentation, Measurement, Circuits and Systems*, 2011, pp. 57–60.

[83] N. Zazworka, K. Stapel, E. Knauss, F. Shull, V. R. Basili, and K. Schneider, "Are developers complying with the process: An XP study," in *Proceedings of the 4th International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. Bolzano-Bozen, Italy: IEEE Computer Society, Sept. 2010.

[84] H. Zhang, "Special panel: Software process simulation — at a crossroads?" in *Proceedings of the International Conference on Software and System Process (ICSSP)*. IEEE, 2012, pp. 215–216.

[85] H. Zhang, R. Jeffery, D. Houston, L. Huang, and L. Zhu, "Impact of process simulation on software practice: an initial report," in *Proceedings of the 33rd International Conference on Software Engineering (ICSE)*, 2011, pp. 1046–1056.

[86] H. Zhang, B. Kitchenham, and D. Pfahl, "Software process simulation modeling: an extended systematic review," in *Proceedings of the International Conference on Software Process (ICSP)*. Springer, 2010, pp. 309–320.

[87] ——, "Software process simulation over the past decade: trends discovery from a systematic review," in *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Kaiserslautern, Germany: ACM, 2008, pp. 345–347.

Chapter 2

A systematic literature review on the industrial use of software process simulation

Abstract

Software process simulation modeling (SPSM) captures the dynamic behavior and uncertainty in the software process. Existing literature has conflicting claims about its practical usefulness: SPSM is useful and has an industrial impact; SPSM is useful and has no industrial impact yet; SPSM is not useful and has little potential for industry. To assess these conflicting standpoints on the usefulness of SPSM. A systematic literature review was performed to identify, assess and aggregate empirical evidence on the usefulness of SPSM. In the primary studies, to date, the persistent trend is that of proof-of-concept applications of software process simulation for various purposes (e.g. estimation, training, process improvement, etc.). They score poorly on the stated quality criteria. Also, only a few studies report some initial evaluation of the simulation models for the intended purposes. There is a lack of conclusive evidence to substantiate the claimed usefulness of SPSM for any of the intended purposes. A few studies that report the cost of applying simulation do not support the claim that it is an inexpensive method. Furthermore, there is a paramount need for improvement in conducting and reporting simulation studies with an emphasis on evaluation against the intended purpose.

2.1 Introduction

Delivering high quality software products within resource and time constraints is an important goal for the software industry. An improved development process is seen as a key to reach this goal. Both academia and industry are striving to find ways for continuous software process improvement (SPI). There are numerous SPI frameworks and methodologies available today [7, 47, 17], but they all have one challenge in common: the cost of experimenting with the process change. It is widely claimed that software process simulation modeling (SPSM) can help in predicting the benefits and repercussions of a process change [40], thus, enabling organizations to make more informed decisions and reduce the likelihood of failed SPI initiatives.

Since the suggestion to use simulation modeling for understanding the software development process by McCall et al. [108] in 1979, there is considerable literature published over the last three decades in this area. There are a number of secondary studies on the subject that have scoped the research available on the topic [19, 27, 57, 58, 59, 56, 14].

From these studies, it can be seen that all SPSM purposes identified by Kellner et al. [19] have been explored in SPSM research over the years. In terms of the scope of the simulation models, it has ranged from modeling a single phase of the life-cycle to various releases of multiple products [57, 56]. The following is a brief list of some proclaimed benefits of SPSM:

- Improved effort and cost estimation
- Improved reliability predictions
- Improved resource allocation
- Risk assessment
- Studying success factors for global software development
- Technology evaluation and adoption
- Training and learning

Such range of claimed potential benefits and reports of industrial application and impact [53] give an impression that simulation is a panacea for problems in Software Engineering (SE). However, some authors have recently questioned the validity of these claims [37]. Three positions can be delineated from literature on SPSM:

Claim 1: software process simulation is useful in SE practice and has had an industrial impact [53].

Claim 2: SPSM is useful however it is yet to have a significant industrial impact [28, 16, 8].

Claim 3: questions not only the usefulness but also the likelihood and potential of being useful for the software industry [37].

In this study, we aim to aggregate and evaluate, through a systematic literature review [24], the empirical evidence on the usefulness of SPSM in real-world settings (industry and open source software development). In essence, we aim to establish which of the claims in the SPSM community can be substantiated with evidence. The main contributions of this study can be summarized as the following:

- We attempted to substantiate the claim that SPSM is an inexpensive [110, 19] mechanism to assess the likely outcome before actually committing resources for a given change in the development process [19].
- We attempted to characterize which SPSM approaches are useful for what purpose and under which context in real-world software development (cf. [19] for a definition of purpose and scope).
- We used a systematic and documented process to identify, evaluate, and aggregate the evidence reported for the usefulness of SPSM [24, 18].
- The existing secondary studies cover literature published from 1998 till December 2008. We included any literature published till December 2012 and also considered other than typical SPSM venues and found substantially more studies (a total of 87 primary studies of which 17 are published before 1998, 46 between 1998 and 2008, and 24 after 2008) that have used SPSM in a real-world software development setting than any of the existing secondary studies.
- From the existing secondary studies, we now know that many simulation approaches *"can be applied"* and that they *"can be useful"*. However, in this study we attempt to see if there is a progression in SPSM literature and if these claims can now be substantiated.
- By following an objective, thorough and systematic approach (detailed in Section 2.3) the existing research on SPSM is evaluated in an objective, unbiased manner. This well-intentioned endeavor is to identify improvement opportunities to raise the quality and steer the direction of future research.

The remainder of the chapter is structured as follows: Section 2.2 presents the related work. Section 2.3 explains our research methodology. Section 2.4 shows the characteristics of the primary studies, followed by the review results in Section 2.5. Section 2.6 discusses the results of the systematic literature review, and Section 2.7 concludes the chapter.

2.2 Related work

Using the search strategy reported in Section 2.3.2, we identified a number of existing reviews of the SPSM literature [19, 27, 57, 58, 59, 56, 14, 5]. These are mostly mapping studies that provide an overview of the SPSM research. None of these studies help to assess which of the claims about the usefulness of SPSM are backed by evidence.

Kitchenham and Charters [24] have identified criteria to assess an existing review. From these criteria, we used the detailed check-list proposed by Khan et al. [20] and the general questions recommended by Slawson [45]. The aim was to evaluate the existing reviews on their objectives, coverage (data sources utilized, restrictions etc.), methodology, data extraction, quality assessment, analysis, validity and reporting. The detailed criteria are available in [4].

2.2.1 Existing reviews

The results of our assessment using these criteria [4] on the existing literature reviews in the SPSM field are presented in the following subsections.

Kellner et al. [19]

Kellner et al. [19] provide an overview of SPSM field and identify the objectives for use of simulation, scope of simulation models and provide guidance in selecting an appropriate modeling approach. They also summarize the papers from the First International Silver Falls Workshop on Software Process Simulation Modeling (ProSim'98).

Their study was published in 1999 and there is considerable new literature available on the topic. We utilize their work to explore how the research in real-world application of SPSM has used the simulation approaches for the purposes and scopes identified in their study.

Zhang et al. [57, 58, 59, 56]

Zhang et al. [57, 58, 59, 56] reported a *"two-phase"* scoping study on SPSM research. They have used six broad questions to scope [35] the field of SPSM. They [56] also acknowledge that their study *"is also a kind of mapping study"*.

In the initial phase, Zhang et al. [57] performed a manual search of renowned venues for SPSM literature. In the second phase [56], it was complemented with an electronic search in IEEE, Science Direct, Springer Link and ACM, covering literature from 1998-2007.

The use of only one reviewer for selection, data extraction, quality assessment and study categorization is a potential threat to the validity of their studies. With such a large amount of literature that one reviewer had to go through for these two broad studies, a reviewer is highly likely to make mistakes or overlook important information [22, 51]. If only one reviewer is doing the selection there is no safety net and any mistake can result in missing out a relevant article [21]. Another shortcoming, as acknowledged by the authors is the use of a less rigorous process for conducting the review [57], *"the main limitation of our review is that the process recommended for PhD candidates is not as rigorous as that adopted by multiple-researchers"*.

For the tasks where a second reviewer (e.g. for data extraction from 15% of the studies in the second phase [56]) was involved, neither the inter-rater agreement nor the mechanism for resolution of disagreements is described.

Liu et al. [27]

Liu et al. [27] primarily scoped the research on software risk management using SPSM. They seek answers for five broad scoping questions but focusing on use of SPSM in software risk management. The mapping results represent the studied purposes, the scope of the modeled processes and the tools used in the primary studies.

They used the same electronic databases as Zhang et al. [57, 56] for automatic search and also manually traversed the proceedings of Software Process Simulation and Modeling Workshop (ProSim) (1998-2006), International Conference on Software Process (2007-2008), Journal of Software Process Improvement and Practice (1996-2007) and special issues of Journal of Systems and Software Volume 46, Issues 2-3, 1999 and Volume 59, Issue 3, 2001.

Like the Zhang et al. study [57, 56], Liu et al. [27] did not use the quality assessment results in the selection of studies or in the analysis. Their entire review was done by one reviewer *"One PhD student acted as the principal reviewer, who was responsible for developing the review protocol, searching and selecting primary studies, assessing the quality of primary studies, extracting and synthesizing data, and reporting the review results."*

Zhang et al. [53]

Zhang et al. [53] present an overview of software process simulation and a historical account/time-line of SPSM research, capturing who did what and when. They claim to have done some impact analysis of SPSM research based on the results of their earlier reviews [57, 58, 59, 56]. The *"case study"* reported in this article to supplement the

"impact" analysis is at best anecdotal and is based on *"interview-styled email communications"* with Dr. Dan Houston. They have acknowledged this to be an initial study that needs to be extended when they say *"we are fully aware that our results are based on the examination of a limited number of cases in this initial report. The impact analysis will be extended to more application cases and reported to the community in the near future"*.

Furthermore, the following conclusions in the article [53] are not backed by traceable evidence reported in primary studies included in their review:

- *"It is shown that research has a significant impact on practice in the area"* i.e. SPSM in practice.
- *"Anecdotal evidence exists for the successful applications of process simulation in software companies"*.
- *"The development of an initial process simulation model may be expensive. However, in the long-term, a configurable model structure and regular model maintenance or update turn out to be more cost effective"*.

de França and Travassos [14]

de França and Travassos [14] characterized the simulation models in terms of model type, structure, verification and validation procedures, output analysis techniques and how the results of the simulation were presented in terms of visualization. Their study is different from previously discussed reviews (in sections Section 2.2.1-Section 2.2.1) as it considers verification and validation of models, and hence has an element of judging the quality of the simulation models being investigated. Another difference that is important to note is their inclusion of all simulation studies that were related to the software engineering domain (e.g. architecture) thus covering simulation as a whole. They used Scopus, EI Compendex, Web of Science and developed their search string by defining the population, intervention, comparison, and outcome.

In their selection of studies, one reviewer conducted the selection first, his decisions were reviewed by a second reviewer and lastly the third reviewer cross-checked the selection. This increased the validity of study selection however it is prone to bias as the selection results from the first reviewer were available to the second reviewer and subsequently both the categorizations potentially biased the selection decision of the third reviewer.

The list of primary studies and the results of quality appraisal are reported in the study. They have also reported their data extraction form, but did not report on the measures undertaken to make the data extraction and classification of studies more reliable.

They extracted information about model verification and validation and how many studies conducted this activity in different ways. This provides an interesting point of comparison with our study as both studies conducted this assessment independently without knowing each others outcomes.

Bai et al. [5]

Bai et al. [5] conducted a secondary study of empirical research on software process modeling without an explicit focus on SPSM. The study has four research questions that scope the empirical research in software process modeling for:

- Research objectives
- Software process modeling techniques
- Empirical methods used for investigation
- Rigor of studies (whether research design and execution are reported)

The study used *"7 journals and 6 conference proceedings, plus other relevant studies found by searching across online digital libraries, during the period of 1988 till December 2008"*. Although the general selection criteria and data extraction form are presented, no details of the procedure for selection, extraction or quality evaluation are presented. The detailed criteria for how the rigor of studies was evaluated are also not reported. Likewise, it is unclear what was the role of each reviewer in the study. Without this information it is difficult to judge whether the results are sensitive to the way the review was conducted.

Given that only 43 empirical studies are identified in their review raises some concerns about their search strategy (selection of venues, search strings etc.). Since their study had a broader scope than ours which is including all software process modeling literature, there should have been substantially more studies.

To aggregate results they used frequency analysis in terms of how many studies investigated a specific research objective, process modeling technique, using a certain empirical method and how many described the design and execution of the study.

2.2.2 Our contribution

The contributions of this study in comparison to existing secondary studies can be summarized as following:

1. Given the conflicting claims about the usefulness of SPSM it was important to use a systematic methodology to ensure reliability and repeatability of the review to the extent possible. We have decided to use two reviewers and other

preventive measures (discussed in detail in Sections 2.3.4, 2.3.5, 2.3.7 and 2.3.6) to minimize the threats of excluding a relevant article. These measures included using pilots, inter-rater agreement statistics and a documented process to resolve differences. With these we aimed to reduce the bias in various steps of selection, extraction and analysis of the primary studies.

2. The conflicting positions with regard to SPSM could not be resolved based on the existing secondary studies because their focus is not to identify and aggregate evidence (as discussed in Section 2.2.1). On the contrary, our contribution is the identification of research studies using SPSM in real-world software development followed by an attempt to evaluate and aggregate the evidence reported in them. Thus, investigating if the claims of potential benefits can be backed by evidence.
3. In theory, a systematic literature review is an exhaustive study that evaluates and interprets *"all available research"* relevant to the research question being answered [24]. However, in practice it is a subset of the overall population that is identified and included in a study [51]. In this study, however, we aspired to take the study population as close to the actual population. The number of studies identified in this study compared to other reviews is discussed in detail in Section 2.6.6. To achieve this we took following decisions:

    - Search in not restricted to only typical venues of SPSM publications and includes the databases that cover Computer Science and SE literature.
    - Lastly in the existing reviews, the potentially relevant sources for the management and business literature were not included in the search. Zhang et al. [56] noticed that SPSM research mainly focuses on managerial interests. Therefore, it is highly probable that SPSM studies may be published outside the typical Computer Science and SE venues. Thus, in this literature review, we also searched for relevant literature in data sources covering these subjects. In particular, business source premier was searched that is specifically targeting business literature.
    - The secondary studies by Bai et al. [5], Liu et al. [27] and Zhang et al. [57, 58, 59, 56] only cover literature published between 1998 and 2008, in this systematic literature review we do not have an explicit restriction on the start date and include all literature published till December 2012. Given the noticeable trend of increasing empirical research reported in these studies [5, 27] our study also contributes by aggregating the more recent SPSM literature.
    - No start date was put on the search to exhaustively cover all the literature available in the selected electronic databases up till the search date (i.e.

December 2012). This enabled us to identify the earliest work by McCall from 1979 [108] and also include earlier work of Abdel-Hamid [60, 61, 62, 66, 63, 64, 65].

4. Other secondary reviews only scoped the existing research literature and did not highlight the lack of evaluation of claimed benefits. Overall, in a systematic and traceable manner we identify the limitations of current research in terms of reporting quality, lack of verification and validation of models, and most significantly the need to evaluate the *usefulness* of simulation for different purposes in various contexts.
5. This review by identifying the limitations in the current SPSM research has taken the first step towards improvement. The criticism of SPSM is not intended to dismiss its use, but to identify the weaknesses, raise awareness and hopefully improve SPSM research and practice. We have also provided recommendations and potential directions to overcome these limitations and perhaps improve the chances of SPSM having an impact on practice.

2.3 Research methodology

To identify appropriate SPSM approaches for given contexts and conditions a systematic literature review following the guidelines proposed by Kitchenham et al. [24] was performed. We attempted to aggregate empirical evidence regarding the application of SPSM in a real-world settings.

2.3.1 Review question

To assess strength of evidence for usefulness of simulation in real-world use we attempt to answer the following research question with a systematic literature review:

RQ: *What evidence has been reported that the simulation models achieve their purposes in real-world settings?*

2.3.2 Need for review

As a first step in our review, to identify any existing systematic reviews and to establish the necessity of a systematic review, a search in electronic databases was conducted. The keywords used for this purpose were based on the synonyms of systematic review methodology listed by Biolchini et al. [10] along with *"systematic literature review"*.

The search was conducted in the databases identified in Table 2.1, in year 2013, using the following search string with two blocks joined with a Boolean 'AND' operator:

(software AND process AND simulation) ***AND*** *("systematic review" OR "research review" OR "research synthesis" OR "research integration" OR "systematic overview" OR "systematic research synthesis" OR "integrative research review" OR "integrative review" OR "systematic literature review")*

This search string gave 47 hits in total. After removing duplicates, titles and abstracts of the remaining articles were read. This way we identified five articles that report two systematic reviews [57, 58, 59, 56] and [27].

By reading the titles of articles that cite these reviews, we identified two more relevant review articles [53, 14]. In Section 2.2, we have already discussed in detail the limitations of these articles. We have also discussed the novel contributions of our study and how we have attempted to overcome the shortcomings in these existing reviews.

Table 2.1: Digital databases used in the study

Database	Motivation
IEEE, ACM Digital and Engineering Village (Inspec and Compendex) and Science direct	For coverage of literature published in CS and SE.
Scopus, Business source premier, Web of science	For broader coverage of business and management literature along with CS, SE and related subject areas.
Google Scholar	To supplement the search results and to reduce the threats imposed by the limited search features of some databases this search engine was used.

2.3.3 Search strategy

A conscious decision about the keywords and data-sources was made that is detailed below along with the motivation:

Data sources

Since, the study is focused on the simulation of software development processes, therefore it is safe to look for relevant literature in databases covering Computer Science

(CS) and Software Engineering (SE). However, as the application of simulation techniques for process improvement may be published under the business related literature, e.g. organizational change, we decided to include databases of business literature as well.

Keywords

Starting with the research questions suitable keywords were identified using synonyms, encyclopaedia of SE [29] and seminal articles in the area of simulation [19]. The following keywords were used to formulate the search strings:

- **Population:** Software process or a phase thereof. *Alternative keywords:* Software project, software development process, software testing/maintenance process
- **Intervention:** Simulation. Alternative keywords: simulator, simulate, dynamic model, system dynamics, state based, rule based, Petri net, queuing, scheduling.
- **Context:** Real-world. *Alternative keywords:* empirical, industry, industrial, case study, field study or observational study. Our target population was studies done in industry and we intended to capture any studies done in that context regardless of the research method used. We expected that any experiments that have been performed in industrial settings would still be identified. Yet by not explicitly including experiment as a keyword we managed to, some an extent, disregard studies in a purely academic context.
- **Outcome:** Positive or negative experience from SPSM use. Not used in the search string.

The keywords within a category were joined by using the Boolean operator OR and the three categories were joined using the Boolean operator 'AND'. This was done to target the real-world studies that report experience of applying software process simulation. The following is the resulting search string:

((software 'Proximity Op' process) OR (software 'Proximity Op' project)) ***AND*** *(simulat* OR "dynamic model" OR "system dynamic" OR "state based" OR "rule based" OR "petri net" OR "queuing" OR "scheduling")* ***AND*** *(empirical OR "case study" OR "field study" OR "observational study" OR industr*)*

The proximity operator was used to find more relevant results and yet at the same time allow variations in how different authors may refer to a software development process, e.g. software process, software testing process, etc. However, in the databases that did not correctly handle this operator we resorted to the use of Boolean operator AND instead. The exact search strings used in individual databases can be found in [4].

The search in the databases (see Table 2.1) was restricted to title, abstract and keywords except in Google Scholar where it was only done in the title of the publications (the only other option was to search the full-text). Google Scholar is more of a search engine than a bibliographic database. Therefore, we made a trade-off in getting a broader coverage by using it without the *context* block, yet restricting the search in titles only to keep the number of hits practical for the scope of this study.

Figure 2.1 provides an overview of the search results and selection procedure (discussed in detail in Section 2.3.4) applied in this review to identify and select primary studies.

2.3.4 Study selection criteria and procedure

Before the application of selection criteria (related to the topic of the review) all search results were subjected to the following generic exclusion criteria:

- Published in a non peer reviewed venue e.g. books, Masters/Ph.D. theses, keynotes, tutorials and editorials etc.
- Not available in English language.
- A duplicate (at this stage, we did not consider a conference article's subsequent publication in a journal as duplicate this is handled later on when the full-text of the articles was read).

Where possible, we implemented this in the search strings that were executed in the electronic databases. But since many of the journals and conferences published in primary databases are covered by bibliographic databases, we had a high number of duplicates. Also, in Google Scholar we had no mechanism to keep out grey literature from the search results. Therefore, we had to do this step manually.

After this step, the remaining articles were subjected to three sets of selection criteria preliminary, basic and advanced. As the number of search results is fairly large, for practicality, the preliminary criteria were used to remove the obviously irrelevant articles.

Preliminary Criteria

Preliminary criteria were applied on the titles of the articles, the information about the venue and journal was used to supplement this decision. If the title hinted exclusion of articles but there was a doubt the abstract was read. If it was still unclear the article was included for the next step where more information from the article was read to make a more informed decision.

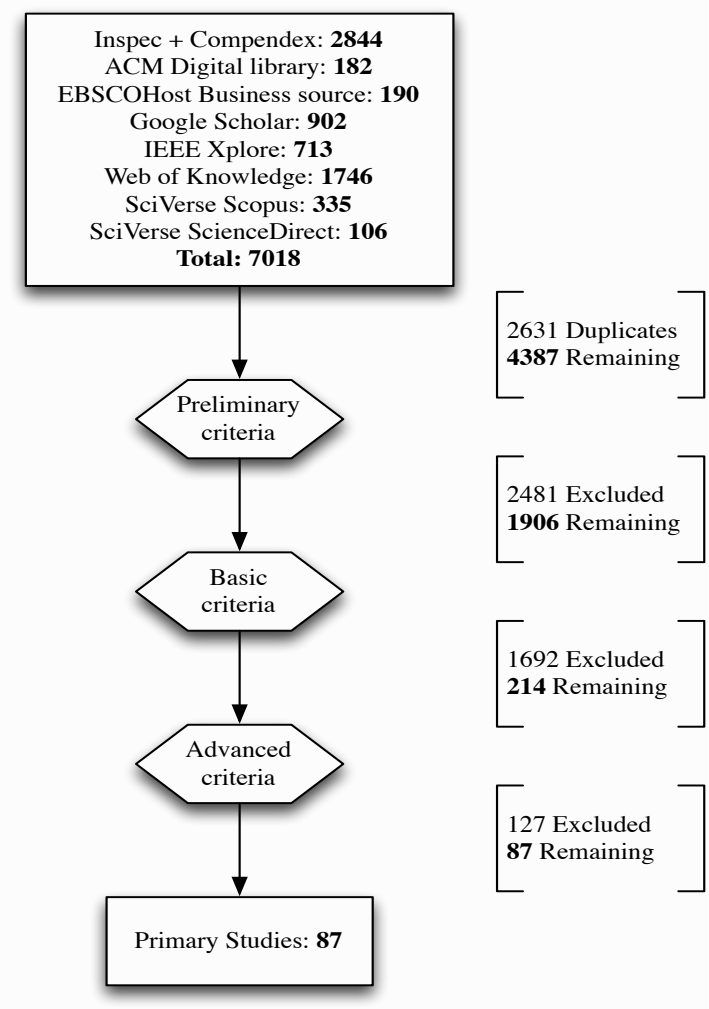


Figure 2.1: Overview of steps in the selection of primary studies.

- Exclude any articles related to the simulation of hardware platforms.
- Exclude any articles related to the use of simulation software, e.g. simulating manufacturing or chemical process or transportation, etc.
- Exclude any articles related to use of simulation for evaluation of software or hardware reliability and performance etc.
- Exclude any articles related to use of simulation in SE education in academia. Articles with educational focus using software process simulation were not rejected straight away based on the title. Instead, we read the abstract to distinguish the SPSM used for training and education in the industry from those in a purely academic context. Only the articles in the latter category were excluded in this study. If such a decision could not be made about the context, the article was included for the next step.

The preliminary criteria were applied by only one reviewer. By using *"when in doubt, include"* as a rule of thumb we ensured inclusiveness to reduce the threat of excluding a relevant article. Also having explicit criteria about what to exclude reduced the reviewer's bias as we tried to minimize the use of authors own subjective judgment in selection.

Basic Criteria

Basic criteria were applied to evaluate the relevance of the studies to the aims of our study by reading the titles and abstracts.

- Include an article related to the simulation of a software project, process or a phase thereof. For example, the type of articles identified by the preliminary criteria.
- Exclude an article that only presents a simulation technique, tool or approach.
- Exclude a non-industrial study (e.g. rejecting the empirical studies with students as subjects or mock data). Studies from both commercial and open source software development domains were included in this review.

It was decided that articles will be labeled as: Relevant, Irrelevant or Uncertain (if available information i.e. title and abstract, is inconclusive). Given that two reviewers will do the selection we had six possibilities (as shown in Table 2.2) of agreement or disagreement between the reviewers about the relevance of individual articles.

In Table 2.2 categories A, C and F are cases of perfect agreement between reviewers. The decision regarding each of the categories motivated by the agreement level of reviewers and likelihood of finding relevant articles in such a category is listed below:

Table 2.2: Different possible scenarios for study selection

		Reviewer 2		
		Relevant	Uncertain	Irrelevant
Reviewer 1	Relevant	A	B	D
	Uncertain	B	C	E
	Irrelevant	D	E	F

- Articles in category A and B (considered potential primary studies) will be directly taken to the last step of full-text reading. Although articles in category B show some disagreement between the authors but (since one author is certain about the relevance and the other is inconclusive) we considered it appropriate to include such studies for full-text reading.
- On the other hand, articles in category F will be excluded from the study as both reviewers agree on their irrelevance.
- Articles in category C will be reviewed further (by both reviewers independently using the steps of adaptive reading described below) where more detail from the article will be used to assist decision making. This was a rational choice to consult more detail, as both reviewers concurred on a lack of information to make a decision.
- Articles in category D and E show disagreement, with category D being the worst as one author considers an article relevant and other considers it irrelevant. Articles in these two categories were deemed as candidates for discussion between reviewers. These articles were discussed and reasons for disagreement were explored. Through consensus, these articles were placed in either category A (included for full-text reading as a potential primary study), C (uncertain need more information and subjected to adaptive reading) or F (excluded from the study).

To develop a common understanding of the criteria both reviewers read the criteria and using *"think aloud"* protocol applied it on three randomly selected articles.

Furthermore, before performing the actual inclusion and exclusion of studies, a pilot selection was performed. This step was done by two reviewers independently on 20 randomly selected articles. The results of this pilot are shown in Table 2.3.

Table 2.3: Results of the pilot selection

		Reviewer 2		
		Relevant	Uncertain	Irrelevant
Reviewer 1	Relevant	6	0	2
	Uncertain	-	4	0
	Irrelevant	-	-	8

We had an agreement on 90% of the 20 articles used in the pilot of inclusion/exclusion criteria. Based on these results with high level of agreement, we were confident to go ahead with the actual selection of the studies.

Inclusion and exclusion criteria were applied independently by two reviewers on 1906 articles that had passed the preliminary criteria used for initial screening (see Figure 2.1). The results of this phase are summarized in Table 2.4 where the third column shows the final total of articles once the articles in category D and E were discussed and reclassified.

Table 2.5 shows good agreement on the outcome of applying basic criteria on articles. This shows a shared understanding and consistent application of the criteria on the articles. Only on 30 out of 1906 articles the reviewers had a major disagreement i.e. category D in Table 2.4.

Adaptive reading for articles in category C:
Based on the titles and abstracts of articles, we often lacked sufficient information to make a judgement about the context and method of the study. Therefore we had 174 articles (category C in Table 2.4) that required more information for decision mak-

Table 2.4: Results of applying the inclusion and exclusion criteria

Category ID	Number of articles	Total number of articles after discussion
A	96	106
B	34	34
C	122	174
D	30	0
E	82	0
F	1542	1592

Table 2.5: Cohen's Kappa and percent agreement between reviewers

Criteria	Percent Agreement	Cohen's Kappa statistic
Basic criteria	92.50	0.73
Adaptive reading	78.60	0.53
Context description	80.50	0.65
Study design description	81.60	0.56
Validity threats discussion	95.40	0.73
Subjects/Users	92.00	0.34
Scale	80.50	0.58
Model validity	89.70	0.83

ing. Many of the existing literature reviews exclude such articles where both reviewers do not consider a study relevant (see Chapter 7) However, as we decided to be more inclusive to minimize the threat of excluding relevant research we decided to further investigate such studies.

As the number of articles in this category was quite large (174 articles) and we already had a sizable population of potential primary studies (106 and 34 articles in category A and B respectively) we could not justify spending a lot of effort in reading full-text of these articles. Therefore, we agreed on an appropriate level of detail to make a selection decision without having to read the full-text of the article. The resulting three-step process of inclusion and exclusion with increasing degree of detail is:

1. Read the introduction of the article to make a decision.
2. If a decision is not reached read the conclusion of the article.
3. If it is still unclear, search for the keywords and evaluate their usage to describe the context of the study in the article.

Again a pilot of this process was applied independently by the two reviewers on five randomly selected articles in category C. The reviewers logged their decisions and the step at which they took the decision e.g. *'Reviewer-1 has included article Y after reading its conclusion'*. In this pilot, we had a perfect agreement on four of the five articles with regard to the decision and the step where the decision was made. However, one article resulted in some discussion as reviewers noticed that in this article authors had used terms "empirical" and "example" and this made it unclear whether the study was done in real-world settings. To avoid exclusion of any relevant articles it was decided that such articles that are inconclusive in their use of these terms will be included for full-text reading.

The adaptive reading process described above was applied independently by both reviewers and we had a high congruence on what was considered relevant or irrelevant to the purpose of this study. The inter-rater agreement was fairly high for this step as presented in Table 2.5. All articles with conflicting decision between the two reviewers were taken to the next step for full-text reading. This resulted in another 74 articles for full-text reading in addition to the 140 articles in category A and B see Table 2.4.

Advanced Criteria

This is related to the actual data extraction, where the full-text of the articles was read by both reviewers independently. Exclude articles based on the same criteria used in the previous two steps (see Section 2.3.4 and Section 2.3.4) but this time reading the full-text of the articles. We also excluded the conference articles that have been subsequently extended to journal articles (that are likely to have more details).

For practical reasons these 214 articles were divided equally among the two reviewers to be read in full-text. However, to minimize the threat of excluding a relevant study any article excluded by a reviewer was reviewed by the second reviewer. Section 2.3.7 presents the data extraction form used in this study, the results of the pilot and the actual data extraction performed in this study. The list of excluded studies at this stage are available in [4].

2.3.5 Study quality assessment criteria

The criteria used in this study were adapted from Ivarsson and Gorschek [18] to fit the area of SPSM. We dropped 'research methodology' and 'context' as criteria from the relevance category because we only included the real-world studies in this review. So, these fields were redundant.

Scoring for Rigor

To assess how rigorously a study was done we used the following three sub-criteria:

- *Description of context*
    1. If the description covers at least four of the context facets: product; process; people; practices, tools, techniques; organization and market [36] then the score is '1'.
    2. If the description covers at least two of the context facets then the score is '0.5'.
    3. If less than two facets are described then the score is '0'.

In general, a facet was considered covered if even one of the elements related to a facet is described. The facet "process" was considered fulfilled if a general description of the process or if name of the process model followed in the organization is provided.

- *Study design description*
  1. If the data collection/analysis approach is described to be able to trace the following then the score is '1', which is given a) what information source (roles/number of people/data set) was used to build the model, and b) how the model was calibrated (variable to data-source mapping), and c) how the model was evaluated (evaluation criteria and analysis approach).
  2. If data collection is only partially described (i.e. at least one of the three - a), b), or c) above has been defined) then the score is '0.5'.
  3. If no data collection approach is described then the score is '0' (example: "we got the data from company X").
- *Discussion of validity threats*
  1. If all four types of threats to validity [50] (internal, external, conclusion and construct) are discussed then the score is '1'.
  2. If at least two threats of validity are discussed then the score is '0.5'.
  3. If less than two threats to validity are discussed then the score is '0'.

Scoring of Relevance

The relevance of the studies for the software engineering practice was assessed by the following two sub-criteria: users/subjects and scale:

- *Users/Subjects*
  1. If the intended users are defined and have made use of the simulation results for the purpose specified then the score is '1' (in case of prediction, e.g. a follow-up study or a post-mortem analysis of how it performed was done).
  2. If the intended users are defined and have reflected on the use of the simulation results for the purpose specified then the score is '0.5'
  3. If the intended users have neither reflected nor made practical use of the model result then the score is '0' (e.g. the researcher just presented the result of the simulation and reflected on the output in the article).

- *Scale*
    1. If the simulation process is based on a real-world process then the score is '1' (articles that claim that the industrial process is similar to a standard process model were also scored as '1').
    2. If the simulation process has been defined by researchers without industry input then the score is '0' (the articles that only calibrate a standardized process model, will also get a zero).

To minimize the threat of researchers bias both reviewers performed the quality assessment of all the primary studies independently. Kappa statistic for inter-rater agreement was computed see Table 2.5. Generally we had a fair agreement as shown by the values of Cohen's Kappa (values greater than 0.21 are considered fair agreement). However, for criteria like Subjects/Users where we had a low agreement we do not think it is a threat to the validity of the results as all the conflicts were resolved by discussion and referring back to the full-text of the publication. The results of quality assessment of primary studies after consensus are given in Table 2.6.

Table 2.6: Rigor, relevance and model validity scores for the primary studies

Reference	Context description	Study design description	Validity discussion	Subjects	Scale	Model validity
[71]	0	0	0	0	0	0.5
[76]	0	0	0	0	0	0
[132]	0	0	0	0	0	0.5
[131]	0	0	0	0	0	0
[125]	0	0	0	0	0	0
[99]	0	0	0	0	0	1
[142]	0	0	0	0	0	0.5
[81]	0	0	0	0	0	0
[75]	0	0	0	0	0	0
[146]	0	0	0	0	0	0.5
[74]	0	0	0	0	0	0.5
[84]	0	0	0	0	0	0
[89]	0	0.5	0	0	0	1
[129]	0	0	0.5	0	0	0.5
[91]	0	0.5	0	0	0	0.5

Continued on next page

Table 2.6 – *Continued from previous page*

Reference	Context description	Study design description	Validity discussion	Subjects	Scale	Model validity
[97]	0	0.5	0	0	0	0
[86]	0	0	0.5	0	0	0.5
[83]	0	0.5	0	0	0	0.5
[106]	0	0	0	0	1	0.5
[120]	0	0	0	0	1	0.5
[112]	0	0	0	0	1	0
[124]	0	0	0	0	1	0
[87]	0	0	0	0	1	0
[130]	0	0	0	0	1	1
[140]	0	0	0	0	1	0
[103]	0	0	0	0	1	0.5
[117]	0	0	0	0	1	0
[110]	0	0	0	0	1	1
[123]	0	0	0	0	1	0.5
[70]	0	0	0	0	1	1
[73]	0	0	0	0	1	0
[109]	0	0	0	0	1	0
[141]	0	0	0	0	1	1
[137]	0	0	0	0	1	0.5
[121]	0	0	0	0	1	0
[144]	0	0	0	0	1	0.5
[143]	0	0	0	0	1	0.5
[139]	0	0	0	0	1	1
[82]	0	0	0	0	1	0
[118]	0	0.5	0	0	1	1
[105]	0	0	0	0	1	0.5
[111]	0	0.5	0	0	1	0.5
[119]	0	0.5	0	0	1	1
[94]	0	0.5	0	0	1	1
[67]	0	0.5	0.5	0	1	0
[66]	0	0.5	0	0	1	0.5
[92]	0	0.5	0.5	0	1	1
[96]	0	0.5	0	0.5	1	0

Continued on next page

Table 2.6 – *Continued from previous page*

Reference	Context description	Study design description	Validity discussion	Subjects	Scale	Model validity
[95]	0	0.5	0	0.5	1	0
[100]	0.5	0	0	0	0	0.5
[138]	0.5	0	0	0	0	0.5
[80]	0.5	0	0	0	0	0
[102]	0.5	0	0	0	0	0
[126]	0.5	0	0	0	0	1
[134]	0.5	0	0	0	0	0
[107]	0.5	0.5	0	0	0	0.5
[78]	0.5	0	0	0	1	0.5
[63]	0.5	0	0	0	1	0.5
[65]	0.5	0	0	0	1	0.5
[61]	0.5	0	0	0	1	0.5
[64]	0.5	0	0	0	1	0.5
[108]	0.5	0	0	0	1	0.5
[62]	0.5	0	0	0	1	0.5
[114]	0.5	0	0	0	1	0
[113]	0.5	0	0	0	1	0.5
[115]	0.5	0	0	0	1	0.5
[128]	0.5	0	0	0	1	0.5
[69]	0.5	0	0	0	1	0.5
[122]	0.5	0	0	0	1	0.5
[79]	0.5	0.5	0	0	1	1
[133]	0.5	0.5	0	0	1	0.5
[116]	0.5	0.5	0	0	1	0
[88]	0.5	0.5	0	0	1	0.5
[85]	0.5	0.5	0.5	0	1	0.5
[136]	0.5	1	0	0	1	0.5
[93]	0.5	0	0	1	1	0.5
[68]	0.5	0.5	0.5	1	1	0
[145]	0.5	0	0	0	0	0
[101]	0.5	0	0	0	1	0.5
[98]	1	0.5	0.5	0	0	0.5
[135]	1	0	0	0	1	0.5

Continued on next page

Table 2.6 – *Continued from previous page*

Reference	Context description	Study design description	Validity discussion	Subjects	Scale	Model validity
[60]	1	0	0	0	1	0.5
[72]	1	0.5	0	0	1	0.5
[104]	1	0.5	0	0	1	1
[127]	1	0.5	0	0	1	1
[77]	1	0.5	0	0	1	1
[90]	1	1	0.5	0	1	0.5

2.3.6 Scoring model validity

To assess the credibility of models (see Chapter 4 for details of measures to achieve it) developed and applied in the primary studies we used the following criteria:

1. If the following two steps were performed the model was scored as '1': a) The model was presented to practitioners to check if it reflects their perception of how the process works [19, 42], or did sensitivity analysis[19, 30]; b) Checked the model against reference behavior [44, 30] or compared model output with past data [2] or show model output to practitioners.
2. If at least one of a) or b) is reported then the score is '0.5'.
3. If there is no discussion of model verification and validation (V&V) then the score is '0'.

Both reviewers applied these criteria independently on all the primary studies. Cohen's Kappa value for inter-rater agreement for *"Model Validity"* is 0.83 (Table 2.5). This shows a high agreement between the reviewers and reliability of this assessment is also complemented by resolving all the disagreements by discussion and referring back to full-text of the publications.

2.3.7 Data extraction strategy

We used a random sample of 10 articles from the selected primary studies for piloting the data extraction form. The results were compared and discussed, this helped in developing a common interpretation of the fields in the data extraction form. This pilot also served to establish the usability of the form whether we did find the relevant information at all in the articles. The data extraction form had the following fields:

- ***Meta information:*** Study ID, author name, and title and year of publication.
- ***Final decision:*** Excluded if a study does not fulfil advanced criteria presented in Section 2.3.4.
- ***Quality assessment:*** Rigor (context description, study design description, validity discussion) and relevance(subjects/users, scale).
- ***Model building:*** Problem formulation (stakeholders, scope and purpose), simulation approach and tools used, data collection methods, model implementation, model verification and validation, model building cost, level of reuse, evaluation for usefulness, criteria and outcome of evaluation, documentation, and benefits and limitations.
- ***Reviewer's own reflections***: The reviewers document notes, e.g. if an article has an interesting point that can be raised in the discussion.

We aimed to identify the intended purpose in the study as stated by their authors and not the potential/possible use of the simulation model in the study. In this regard, using the purpose statements extracted from the primary studies we followed the following three steps to aggregate the repeating purposes in the primary studies:

- **Step-1:** Starting with the first purpose statement create and log a code.
- **Step-2:** For each subsequent purpose statement identify if a purpose already exists. If it does log the statement with the existing code, otherwise create a new code.
- **Step-3:** Repeat Step-2 until the last statement has been catalogued.

The resulting clusters with same coded purpose were mapped to purpose categories defined in [19]. However, we found that the purpose category "Understanding" overlaps with training and learning and it is so generic that it could be true for any simulation study no matter what was the purpose of the study.

Traceability was ensured between the mapping, clusters, and the purpose statements extracted from the primary studies. This enabled the second reviewer to review the results of the process above whether the statements were correctly clustered together. Any disagreements between the reviewers regarding the classification were resolved by discussion.

Similarly, the descriptive statements regarding the simulation model's scope that were extracted from the primary studies were analyzed and mapped to the scopes identified by Kellner et al. [19]. This mapping was also reviewed by the second author for all the primary studies.

2.3.8 Validity threats

The threat of missing literature was reduced by using databases that cover computer science and software engineering. We further minimized the threat of not covering the population of relevant literature by doing a search in databases covering management related literature. Another step to ensure wide coverage was to consider all literature published before the year 2013 in this study. Thus, the search was not restricted by the time of publication or venue in any database.

Using the *"context"* block in the search string (as described in Section 2.3.3) adds a potential limitation to our search approach i.e. the articles that mention the name of the companies instead of the identified keywords, will not be found although they are industrial studies. However, this was a conscious decision as most often applied research is listed with the keywords used in this block. This was also alleviated to some an extent by using a broader search string in Google-Scholar as described in Section 2.3.3. By using an electronic search (with search string) we reduced the selection bias (of reviewers) as well.

For practical reasons, the preliminary criteria were applied by one reviewer that may limit the credibility of the selection. However, by only removing the obviously outside the domain articles which were guided by simple and explicit criteria we tried to reduce this threat. Furthermore, at this stage and the later stages of selection we were always inclusive when faced with any level of uncertainty, this we consider also minimized the threat of excluding a relevant article. The selection of articles based on the basic criteria was done by both reviewers and an explicit strategy based on work presented in Chapter 7 was employed.

All the selection of studies, data extraction procedures and quality assessment criteria were piloted and the results are presented in the chapter. Any differences in pilots and actual execution of the studies were discussed and if needed the documentation of the criteria was updated based on the discussions. This was done to achieve consistency in the application of the criteria and to minimize the threat of misunderstanding by either of the reviewers. By making the criteria and procedure explicit, we have minimized the reviewer's bias and dependence of review results on personal judgements. This has further increased the repeatability of the review.

Inter-rater agreement was also calculated for such activities and is discussed in the chapter, where two reviewers performed a task e.g. application of basic criteria for selection and quality assessment. The inter-rater statistics reported in this study generally show a good agreement between reviewers. This shows that the criteria are explicit enough and support replication otherwise we would have had more disagreements. All the conflicts were resolved by discussion and reviewing the primary studies together. This means that even on the criteria where the reviewers had a lower level of agreement

it is not a threat to the results of the study. However, it does point out that the criteria were not explicit enough and is a threat to repeatability of the review.

Studies are in some cases based on Ph.D. theses, e.g. [60, 61]. We evaluated the rigor and relevance based on what has been reported in the article, hence few studies that were based on the theses could potentially score higher. That is, the authors could have followed the step, but due to page restrictions did not report on the results. However, some of the studies only used calibration data and not the model itself. Given this situation, a few individual rigor and relevance scores could change, however, the principle conclusion would not be different.

Study selection and data analysis that resulted in classification of purpose and scope for the models also involved two reviewers to increase the reliability of the review. Explicit definition of criteria, and the experience of reviewers in empirical research in general and simulation in particular also increases the credibility of the review. Kitchenham et al. [23] highlight the importance of research expertise in the area of review to increase the quality of study selection.

2.4 Characteristics of studies

2.4.1 Number of new studies

Contrary to earlier research we identified significantly more studies from the real-world software development context. Zhang et al. [53] stated that they found *"32 industrial application cases"* of which *"given the limited space, this paper, as an initial report, only describes some of the important SPS application cases we identified"*. Similarly, in a systematic literature review of software process modeling literature that included both static and dynamic modeling they found a combined total of only 43 articles [5].

2.4.2 Purpose

Table 2.7 gives an overview of real-world simulation studies relating to the purposes defined in [19]. The clear majority of studies used simulation for planning purposes (45 studies), followed by process improvement (26 studies), and training and learning (21 studies). In comparison, only a few studies used simulation for control and operational management.

Planning: In planning simulation has been used for decision support [65, 135, 96, 90, 87, 129, 62, 60, 132, 109] (e.g. in relation to staffing and allocation of effort [132, 62, 90], connecting decisions and simulating them in relation to business outcomes [87], and requirements selection [129]). Furthermore, simulation has been used

Table 2.7: Purpose of simulation in the primary studies

Purpose	References	Total
Control and Operational Management	[108, 95, 72, 144, 143, 118, 145, 126, 116]	9
Planning	[63, 65, 64, 106, 112, 100, 87, 71, 135, 138, 111, 140, 108, 96, 103, 90, 132, 117, 92, 62, 114, 113, 60, 89, 119, 94, 109, 142, 67, 118, 105, 129, 133, 75, 102, 74, 84, 136, 98, 97, 86, 88, 83, 66, 68]	45
Process Improvement and Technology Adoption	[78, 120, 124, 93, 107, 110, 123, 89, 131, 73, 125, 115, 104, 141, 137, 81, 121, 139, 101, 85, 127, 69, 77, 134, 91, 122]	26
Training and Learning	[61, 130, 76, 95, 117, 92, 70, 89, 131, 99, 128, 82, 80, 79, 92, 133, 75, 146, 127, 69, 77]	21

by many studies for estimation [71, 117, 63, 108, 75, 114, 112, 119, 106, 111, 103, 138, 113, 75, 92, 142, 133, 67], some examples are cost and effort estimation [108, 75, 111], schedule estimation [63], release planning, and fault/quality estimation [138, 106, 100].

Process Improvement and Technology Adoption: Studies in this category used simulation to evaluate alternative process designs for process improvements [107, 121, 137, 120, 124, 141, 93, 104, 101]. As an example, [137] investigated the effect of conducting an improvement plan driven by the ISO/IEC 15504 standard. Furthermore, improvements have been evaluated by varying a number of parameters in the process to determine the best alternatives one ought to strive for (cf. [141, 93, 104]), as well as investigating specific technology adoptions to the process [123, 85, 125, 110, 139, 73, 81]. Examples of technology adoptions were introduction of test driven development (TDD) [110, 139], comparison of manual vs. model-based techniques [73], or use of different quality assurance approaches or architectural styles [81].

Training and Learning: In training and learning the majority of the studies aimed to explore or understand a phenomena from a scientific point of view to provide some recommendations and guidelines to practitioners (cf. [130, 92, 89, 117, 146, 133, 92, 99, 61, 131, 79, 75]). Examples are to understand the effect of requirements overload on bottlenecks in the whole process [92] or understanding open source system development [99], assess what factors make global software development successful [133], or understanding the effects of creeping requirements [117].

Control and Operational Management: Only few studies used simulation for control and operational management [95, 118, 72, 143, 145, 108, 144]. As an example, several studies assessed whether a project is likely to meet its expected deadline [72, 143], or whether an individual iteration is able to meet the deadline [145]. Studies [108, 144] used simulation for progress status assessment.

2.4.3 Scope

Table 2.8 defines categories for scope that a simulation study can have. The clear majority of studies focused on individual projects (56 studies). In comparison, fewer studies looked at the portion of a life-cycle (21 studies), such as testing. Only a small subset of studies investigated simulations in the context of long term evolution (4 studies), long term organization (3 studies), and concurrent projects (3 studies).

Projects: Studies investigating projects and their development life-cycle looked at different life-cycles, e.g. related to CMMI processes [95], and extreme programming (XP) projects [110]. One study explicitly stated that they are focusing on new software development [132], while others in connection with the software life-cycle (excluding requirements and maintenance phase.), see e.g. [63, 61]). The remaining studies only specified that they are looking at the overall development life-cycle of projects without further specification of the type and scope/boundaries.

A Portion of the life-cycle: Studies looking at a portion of a life-cycle investigated focused on the maintenance process solely [78, 121, 128], software reliability life cycle [138], requirements and test [70], quality assurance processes [73, 125], require-

Table 2.8: Scope of simulation models in the primary studies

Scope	**References**	**Total**
A portion of life-cycle	[78, 138, 93, 72, 70, 131, 73, 125, 121, 128, 67, 82, 92, 102, 74, 134, 136, 69, 116, 86, 68]	21
Development project	[63, 61, 64, 106, 112, 100, 124, 71, 135, 130, 111, 140, 108, 107, 96, 95, 103, 90, 132, 117, 110, 62, 114, 113, 123, 60, 89, 119, 99, 115, 104, 94, 109, 142, 81, 144, 143, 139, 80, 118, 101, 105, 145, 133, 85, 75, 126, 84, 98, 91, 77, 97, 88, 122, 83, 66]	56
Concurrent projects	[65, 137, 127]	3
Long term evolution	[141, 79, 129, 146]	4
Long term organization	[120, 87, 76]	3

ments [82], release processes [92], and processes for components off the shelf (COTS) selection and use [131].

Long term product evolution: Researchers have looked at the general long-term evolution of products without more specific classification [141, 79, 146] while [129] looked at market-driven requirements processes from a long-term perspective.

Long term organization: From an organizational perspective, studies investigated factors influencing process leadership [120], processes at an organizational level [87], and CMMI process in relation to organizational business concerns [76].

Concurrent Projects: Two parallel projects have been simulated by [65], while multiple concurrent projects have been simulated by [137] and [127].

2.4.4 Simulation approaches

Table 2.9 shows the simulation approaches used by the identified studies. System dynamics (SD) is the most commonly used simulation approach (32 studies), followed by discrete event simulation (DES) with 22 studies. A total of 12 studies combined different simulation approaches. Only few studies used approaches such as Petri nets (PN) (6 studies) and Monte Carlo (3 studies).

Hybrid simulation models were mostly combining SD and DES [107, 117, 110, 82, 145, 133, 75]. Furthermore, qualitative simulation was combined with other models such as SD, or used to abstract from a quantitative model (cf. [146, 128]). Furthermore, models combined stochastic and deterministic aspects in a single model [89].

Others include a variety of models that used approaches from control theory [142], Copula methods [143], Markov Chaims [85], agent-based [80, 136, 69], while others

Table 2.9: Simulation approaches used in the primary studies

Approach	References	Total
DES	[78, 112, 138, 93, 111, 140, 76, 72, 123, 125, 121, 67, 92, 129, 124, 115, 102, 126, 134, 98, 86, 122, 87]	23
Hybrid	[107, 117, 110, 89, 128, 82, 145, 133, 75, 146, 83, 68]	12
Monte Carlo	[90, 81, 84]	3
Other	[142, 143, 85, 80, 108, 73, 95, 136, 69, 97, 91]	11
PN	[100, 135, 96, 114, 113, 101]	6
SD	[63, 65, 61, 64, 106, 120, 71, 130, 103, 132, 62, 70, 60, 131, 119, 99, 104, 94, 109, 141, 137, 144, 139, 118, 105, 79, 74, 127, 77, 116, 88, 66]	32

did not specify the approach used, and we could not deduce the approach from the information presented.

In connection to simulation approaches, it is interesting which tools have been used to implement them.

SD: Three studies reported the use of Vensim [131, 119, 118]. Other simulation tools used were Powersim [70], and iThink [106]. One study used a self-developed tool [105]. The majority of SD studies did not report the tool they used.

DES: For DES two studies used Extend [125, 67] and two studies used Process Analysis Tradeoff Tool (PATT) [112, 111]. Further tools used were Micro Saint [78], RSQsim [129], Telelogic SDL modeling tool [92] and SoftRel [138]. For two studies simulation models were programmed from scratch in Java [76] and C++ [72]. One study used Statemate Magnum [124] and Nakatani [115] did not specify the tool used.

Hybrid: For hybrid simulation a variety of tools have been used, such as DEVSim++ (APMS) [117], Extend [82], iThink [89], Little-JIT [75], QSIM [128], and Vensim [146]. One study used a self-developed model written in Smalltalk [110].

PN: One study documented that they developed the model from scratch using C [100], others did not report on the tools used.

Monte Carlo: No information about tools has been provided.

2.4.5 Cross analysis purpose and scope

Table 2.10 shows a cross-analysis of purpose and scope. The simulation of individual development projects is well covered across all purposes with multiple studies for each purposes.

A portion of the life-cycle was primarily investigated for process improvement and technology adoption (8 studies) and planning (8 studies). A few studies investigated training and learning (6 studies), and only two studies focused on control and operational management.

Overall, research on concurrent projects is scarce. One study investigated concurrent projects for planning, two for process improvement and technology adoption, and one for training and learning.

Similar patterns are found for long term evolution and long term organization, where primarily individual studies investigated the different purposes with respect to the defined scopes of the simulation models.

No studies in the area of control and operational management focus on long term evolution or organization, which is by definition to be expected.

Table 2.10: Purpose and scope of SPSM in the primary studies

	Control and Operational Management	Planning	Process Improvement and Technology Adoption	Training and Learning
A portion of life-cycle	[72, 116]	[138, 92, 68, 74, 86, 102, 136]	[78, 93, 131, 73, 125, 121, 69, 134]	[70, 131, 128, 82, 92, 69]
Development project	[108, 95, 144, 143, 118, 145, 126]	[63, 64, 106, 112, 100, 71, 135, 111, 140, 108, 96, 103, 90, 132, 117, 62, 114, 113, 60, 89, 119, 94, 109, 142, 118, 105, 133, 75, 66, 83, 84, 88, 97, 98, 67]	[124, 107, 110, 123, 89, 115, 104, 81, 139, 101, 85, 77, 91, 122]	[61, 130, 95, 117, 89, 99, 80, 133, 75, 77]
Concurrent projects		[65]	[137, 127]	[127]
Long term evolution		[129]	[141]	[79, 146]
Long term organization		[87]	[120]	[76]

2.4.6 Cross analysis purpose and simulation approaches

Table 2.11 shows a cross-analysis of purpose and simulation approaches. SD has been used for all purposes, while an emphasis is given to planning (17 studies), followed by process improvement (8 studies) and training and learning (8 studies). Only three SD studies focused on control and operational management.

Hybrid simulation has been used for all purposes as well, with the majority of studies focusing on training and learning (7 studies), followed by planning (6 studies), and process improvement (3 studies). Only one hybrid study has been used for control and operational management.

Overall, the number of Monte-Carlo simulations has been low with only three studies. Out of these three, two studies focus on planning and one on process improvement.

Table 2.11: Purpose and approach of SPSM in the primary studies

	Control and Operational Management	Planning	Process Improvement and Technology Adoption	Training and Learning
DES	[72, 126]	[112, 138, 111, 140, 67, 92, 129, 86, 98, 102, 87]	[78, 93, 123, 125, 121, 122, 134, 124, 115]	[76, 92]
Hybrid	[145]	[117, 89, 133, 75, 83, 68]	[107, 110, 89]	[117, 89, 128, 82, 133, 75, 146]
Monte-Carlo		[90, 84]	[81]	
Other	[108, 95, 143]	[108, 142, 136, 97]	[73, 85, 69, 91]	[95, 80, 69]
PN		[100, 135, 96, 114, 113]	[101]	
SD	[144, 118, 116]	[63, 65, 64, 106, 71, 103, 132, 62, 60, 119, 94, 109, 118, 105, 66, 74, 88]	[120, 131, 104, 141, 137, 139, 77, 127]	[61, 130, 70, 131, 99, 79, 77, 127]

PN studies primarily used the simulation for planning purposes, only one study focused on process improvement. "Other" approaches have been used in a balanced way across purposes.

2.5 Systematic literature review results

To assess the evidence of usefulness of simulation for the proposed purpose we would like to take into account the rigor and relevance of studies, model's credibility (in terms of the level of verification and validation), scope of the model and the real-world context in which it was used.

2.5.1 Context of simulation studies

Petersen and Wohlin [36] defined different facets to describe the industrial context of a study. Each facet contains elements that could be described to characterize the facet, serving as a checklist for researchers during reporting of studies. For this study, a characterization of the following facets was sought: 1) product, 2) processes, 3) people, 4) practices, tools, and techniques, 5) product, 6) organization, and 7) market. Among the primary studies only 9% (8 studies) cover at least four context facets [36] and 56% have described less than two context facets in the articles.

2.5.2 Verification and validation of simulation models

To validate the model structure and representativeness:

- 16% of the studies used *"practitioner feedback"*.
- Only six studies i.e. 7% reported performing sensitivity analysis on the model.
- While 76% studies did not report any such effort.

It can be said that in some of these studies this level of validation of representativeness may not have been necessary since the models were based on standards e.g. the IEEE-12207 (under the premise that the organization uses a similar process).

To validate the model behavior:

- 47% of the studies compared the simulation model output with real data.
- 5% (i.e. four studies) reviewed the model output with practitioners.
- 6% (i.e. five studies) either compared the model output with literature or with other models.
- 40% reported no such effort.

2.5.3 Rigor and relevance of primary studies

Figure 2.2 provides an overview of how the primary studies in this review scored against the rigor-relevance criteria.

From this figure, we can divide the studies into four clear sets as listed below. For example, studies are classified as 'A' (high rigor, low relevance) if they have a rigor score above the median value '1.5' and a relevance score below or equal to the median value '1'.

- 81 Studies classified as 'C': with (Low rigor, Low relevance) of (≤ 1.5, ≤ 1).
- 4 Studies classified as 'B': with (Low rigor, High relevance) of (≤ 1.5, > 1).
- 2 Studies classified as 'A': with (High rigor, Low relevance) of (> 1.5, ≤ 1).
- 0 Studies classified as 'D': with (High rigor, High relevance) of (> 1.5, > 1).

Furthermore, Figure 2.2 shows the median age of the articles in each category. It is visible that the quality of the studies does not seem to exhibit a trend of increase in rigor and relevance scores in relation to newer articles.

2.5.4 Evaluation of simulation for the intended purpose

Only 13% (11 studies) actually reported some sort of evaluation of the model's usefulness for the suggested purpose. 6% of studies compared predicted data to real-data

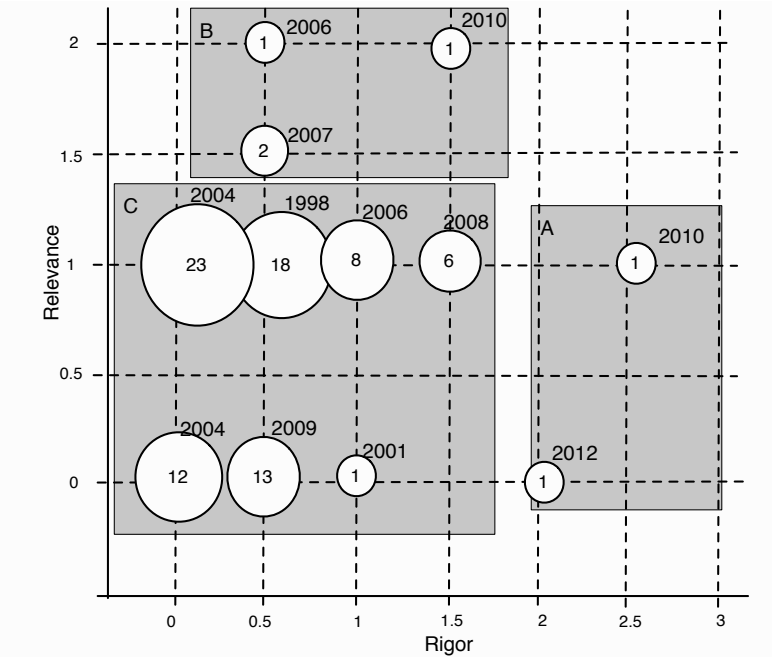


Figure 2.2: Overview of rigor-relevance scores for the primary studies.

(which we have considered an acceptable evaluation provided it is the prediction and not just replication of the real-data based on the calibration of the model). Of the studies 6% used feedback from practitioners for the proposed model purposes (however not based on actual use of the model). Only one study [93] reports a follow-up study where the effectiveness of simulation for the purpose of software process improvement is investigated.

2.6 Discussion

2.6.1 Coverage of modeling scope and purposes

Overall, studies taking a long-term perspective are under-represented, only four studies looked at long term evolution, and three covered long term organization (see Table 2.10). Furthermore, there are only three studies reporting on simulating concurrent

projects, while these become more relevant at this point in time. In particular, in system of systems development and large-scale software development taking a perspective beyond the project is important. From a research perspective, this implies the need for applying simulation on concurrent projects from an end to end development life-cycle. Looking at the intersection between purpose and life-cycle, it becomes more apparent that long-term perspective, long term organization, and concurrent project simulation is a research gap. Only individual studies in those areas focused on planning and process improvement/technology adoption. A cross-analysis of purpose and simulation approaches revealed that overall the different simulation approaches were applied across all the purposes.

2.6.2 Reporting quality

23% of the primary studies partially describe the data collection, analysis and evaluation approach. 62% of these studies do not report the study design in appropriate detail (the criteria was discussed in Section 2.3.5). Only two studies explained the study design in detail. Among the four typical threats to validity [50] in empirical research only 8% of the primary studies had a discussion of two or three threats to validity. Remaining 91% of the primary studies did not have any discussion on validity threats.

This gives a clear picture of the quality of reporting in primary studies and makes it difficult to analyse the credibility and the strength of evidence in these studies. From a secondary studies perspective, this highlights the challenge to synthesize the evidence (if any) reported in these studies.

2.6.3 Context

It is critical to identify the contextual factors because their interaction influences what works well in a certain setting. It is crucial to contextualize empirical evidence by aptly raising the question *"What works for whom, where, when and why?"* [36, 12]. Unfortunately, given that 56% of the articles describe less than two facets of context it is impossible to perform an analysis where we may establish a relation between the usefulness of simulation approaches in a given context using the empirical evidence from the primary studies.

2.6.4 Model validity

Given the lack of empirical data in industrial contexts one would tend to agree with Dickmann et al. [87] when they state that, *"Methodologically, the simplest case is to prove congruence of the simulation results with real world input-output-data. However,*

this type of validation is almost never achieved because of "lack of data" from software development projects". However, surprisingly 51% of the primary studies reported a comparison of the simulation model output with real data for validation.

One reason could be that the goals of the SPSM based study were aligned with the available data. Another explanation could be that the companies where the studies were conducted have extensive measurement programs. The first explanation is highly likely for studies where a phase of the development process is modeled with a narrow focus e.g. the simulation of maintenance process [78]. The second may also hold true as 21% of the studies either simulated a process at NASA or used their data to calibrate the model [60, 61, 62, 63, 64, 65, 112, 111, 123, 125, 104, 81, 82] since they had a strong tradition of measurements.

Overall in terms of V&V of the simulation models built and used in the primary studies, 26% had not reported any V&V whereas 45% had some level of V&V and 16% had reported V&V of both the model's structure and behavior. From the simulation literature [30, 26] we know that the credibility of simulation models cannot be guaranteed by just one of the methods of V&V. This also points to poor credibility of the evidence reported in simulation studies. As pointed out by Ghosh [90] reproducing the output correctly does not validate the underlying cause effect relations in the model. Out of the secondary studies (discussed in Section 2.2.1), only de França and Travassos [14] have critiqued existing research for lacking rigor in model validation and the use of analysis procedures on simulation output data. Our study independently concurs their findings with respect to the model validation [14].

2.6.5 Evaluation of usefulness

Of the 87 primary studies we found a diverse coverage of simulation purpose, model scope and simulation approaches. This was summarized in Section 2.4.5 and Section 2.4.6. However, from the literature review's perspective, the lack of evaluation of simulation models for proposed purposes is disappointing.

Ideally speaking we would like to use the evidence in studies that score high on both rigor and relevance. However, the primary studies in this review scored poorly on both dimensions. Let us therefore look at the studies in categories 'A' and 'B' (that scored high on at least one dimension as reported in Section 2.3.5) that also performed evaluation of simulation models for their intended purposes. We have three such 'B' studies and one 'C' study.

Class 'A':

The two studies [90, 98] though with high rigor yet low relevance scores did not perform an evaluation of the model for the proposed purpose at all.

Class 'B':
Hsueh et al. [95] conducted a questionnaire based survey to assess if the model can be useful for educational purposes. Although they report that most subjects expressed that the model *"could provide interesting and effective process education scenarios"*, but 25% of respondents considered the simulation based games were not helpful or useful to their current work. This can only be considered as an initial evaluation of the approach and given the overall positive outcome a thorough evaluation of the game for educational purpose is required.

Huang et al. [96] report that the project managers claimed that they would have made a value-neutral decision without the (simulation based) analysis from the study. However, in our opinion it is not clear if the contribution is that of the simulation model or that of the underlying *"Value Based Software Quality Achievement"* process framework.

Houston [93] uses DES to quantitatively analyse process improvement alternatives and then performs a follow-up study to see if the changes done in the process resulted in real improvements. Although there is no discussion on how the confounding factors and threats to the validity of the study were mitigated it is difficult to associate the evidence of improvements in the process with use of process simulation to assess improvement alternatives.

Al-Emran [68] combines Monte-Carlo simulation with DES and models the release planning phase. Without presenting any details of the evaluation, they claimed that it was found useful for early risk detection and the development of mitigation strategies in the case study conducted at a company. Furthermore, validation of neither the structure nor the behavior of the model is reported in the article.

The aspect of not performing evaluations may have permeated into SPSM from other disciplines that use simulation, where the emphasis is on verification and validation to ensure that the results are credible. In such disciplines, the results are often considered sufficiently credible if they are accepted and implemented by the sponsor of a simulation study [6]. However, it should be considered that these disciplines have an established tradition of using simulation and ample success-cases exist for them not to do follow-up studies.

Whereas to establish SPSM as an alternative to static process models, analytical models, and as means of planning, management and training in industry, the evidence of its efficacy must be presented. For example in the context of lean development, comparing a value stream mapping workshop [32] with static process descriptions with one facilitated by a dynamic simulation model. Or evaluating the effectiveness of simulation based training of practitioners compared to a well conducted seminar with graphical aids.

Unfortunately, based on the results of this systematic review and the modelers survey [3], it is indicated that SPSM modelers see verification and validation as the evaluation of simulation models. Some initial work towards a framework that highlights the need for evaluation of the value aspect of a simulation model is done by Ahmed et al. [1].

2.6.6 Comparison with previous literature reviews

The total number of simulation studies (87 identified in this study) that were related to real-world application of SPSM is fairly high, which is not indicated by previous literature reviews on simulation. For example, Zhang et al. [53] stated that they found *"32 industrial application cases"* of which *"given the limited space, this paper, as an initial report, only describes some of the important SPS application cases we identified"*. de França and Travassos [14] reported on the frequency of studies with respect to domain, which partially overlaps with what we defined as the scope. A shared observation is that the most frequent investigations relate to development projects. Furthermore, de França and Travassos reported on verification and validation, in particular of 108 of their included studies 17 papers compared their simulation results with actual results, 14 had model results reviewed by experts, and 3 studies used surveys to confirm the validity of model behavior. Whether studies do combinations of them, cannot be deduced from the tables presented. However, the study confirms that only a small portion of studies conducts verification and validation of models. A noteworthy difference between de França and Travassos [14] and the study presented in this chapter is the difference in population and sampling. Their population focused on all types of simulation in software engineering (including architecture simulation), and in their sampling they did not include business literature. Lastly Bai et al. [5] have identified a total of 43 empirical studies even though their population was all software process modeling literature including SPSM.

Furthermore, none of the existing systematic reviews [27, 57, 58, 59, 56] and [5] report the lack of evidence for the usefulness of SPSM in current research. Zhang et al. [53] do indicate a lack of objective evaluation but do not provide any traceable foundation for their claim and they do not highlight the almost non-existence of evidence for the usefulness of SPSM.

2.6.7 Cost of SPSM

When proposing a new tool or practice in industry it is important to not only report the evidence of its effectiveness but also the cost of adoption. Given that simulation

is perceived as an expensive and non-critical project management activity [38] makes reporting the cost of conducting an SPSM study even more important.

However except for two studies none of the primary studies reported the effort spent on simulation. Pfahl and Lebsanft [118] report 18 months of calendar time for the simulation study and an effort of one person year in consulting and 0.25 person years for the development part. In another study, Pfahl [120] only reports the calendar time of three months for knowledge elicitation and modeling and four meetings that were held in this period between Fraunhofer IESE and Daimler Chrysler. Shannon [43] predicted a high cost for simulation as well, *"a practitioner is required to have about 720 hours of formal classroom instruction plus 1440 hours of outside study (more than one man-year of effort)"*.

Given the nature of software development where change is so frequent (technology changes, software development process, environment, and customer requirements etc.), it is very likely that for simulation as a decision support tool will require adaptation. Expecting practitioners to put in one man-year of effort upfront is very unrealistic (especially under the circumstance that we do not have strong evidence for its usefulness). Furthermore apart from the cost in terms of required tool support, training and effort for development, use and maintenance of simulation models there is the cost of the necessary measurement program that can feed the model with accurate data that should also be acknowledged for an effective process simulation.

2.6.8 Accessibility of SPSM studies

In conducting this systematic review, it was pivotal to have access to the Proceedings of the "International Software Process Simulation Modeling Workshop" ProSim. However, these proceedings were not available online e.g. the links (at http://www.icsp-conferences.org/icssp2011/previous.html) and the link to the predecessor Prosim (at http://www.prosim.pdx.edu/) were broken. We obtained the proceedings for ProSim 2003-2005 from a personal contact.

2.6.9 Assessing the evolution of the field

From the point of view of a secondary study that aims to assess and aggregate evidence reported in primary studies, it is very important to understand the contributions of each study. This was particularly difficult in cases of journal articles that were extended from conference articles but did not explicitly acknowledge it, e.g. see the pairs [25] and [100], [89] and [13], [11] and [85], [48] and [139], [106] and [31], as well as [123] and [39]. Furthermore, some studies are remarkably similar, e.g. see the pairs [145]

and [54], [9] and [78], as well as [146] and [55]. A methodology to develop a simulation model proposed by Park et al. is reported in three articles which are very similar as well [117, 33, 34]. Similarly, Zhang et al. have reported a two phase systematic review in five articles [57, 58, 59, 56, 53]. There is an overlap in the contribution of at least three of these articles [57, 58, 59].

From a reporting point of view we therefore recommend to report on reuse of previous models, and extensions made to them to aid other researchers in assessing the contributions made to the field.

In the proceedings of the *"2012 International conference on Software and System Process"* there is an indication that researchers in the SPSM community are trying to ponder on reasons why software process simulation has not had the impact it ought to [28, 38] and [16]. Many of the challenges reported by them are also true in general for all empirical research in SE, e.g. access to practitioners, acquiring credible data, and getting the solutions adopted in industry [46]. Other challenges are perhaps more unique for our research community e.g. SPSM is not sufficiently communicated and is not understood by practitioners [28, 38], and the perceived high cost of SPSM [38]. A more apt reflection on the state of SPSM research is that, *"A retrospective or post-mortem is seldom conducted with SPS so as explain and deepen understanding of a completed project"* [16].

Münch [28] argues that a major challenge in wide spread adoption of SPSM is that the software industry *"is often not ready for applying process simulation in a beneficial way"*. However, based on the results of this systematic literature review, we can see that it will be difficult to build a case for the use of SPSM in practice without reporting the evidence for its usefulness and the cost of adoption.

The results of this systematic literature review corroborate the claim by Pfahl [37] that there is no evidence of wide spread adoption and impact of SPSM research on industry. He also challenges if SPSM has a realistic potential to have an impact on industrial practice [37]. He is not very optimistic about the likelihood of adoption we believe that although he has strong arguments more research is required to make any conclusive statements in this regard. We think that there is a need to identify the problems where the use of simulation can be justified given the high cost of undertaking it and show its utility.

There is a lack of studies evaluating the usefulness of SPSM for training and learning compared to other instructional methods in industrial settings. Pfahl [37] is however more optimistic about the future of SPSM as a means for learning and training in industry. Based on the results of a systematic literature review [49] we do not share the same enthusiasm. Wangenheim and Shull [49] considered any study using a game-based (game or simulation) approach for an educational purpose in their review. The studies included in their review were predominantly based on simulation. They found

that games are effective to reinforce already acquired knowledge. They also concluded that games should only be used in combination with other methods like lectures with an intended learning outcome. Given that they found that games (including simulation based) only have more *"impact on lower cognitive levels, reinforcing knowledge learned earlier"* it is less likely that we will have learning objectives for practitioners that will justify the use of simulation. For example, in Chapter 4, we reported a study where a simulation model was used to illustrate the importance of early integration to the practitioners in a large company that develops software intensive products. However, it is still an open question whether simulation is more effective in getting the message across instead of only static process diagrams, presentations and spreadsheets with graphs showing the potential benefits of early integration and the implications of the current practice.

2.6.10 Recommendations

For practice and research we provide the following recommendations:

1. When using simulation models for scientific purposes (e.g. assessing the usefulness of test driven development), one has to be sure that the appropriate steps with respect to model validity checking have been conducted. Furthermore, given the limitations in demonstrating the usefulness of simulation beyond replication of reference behavior, we recommend to not rely on single simulation studies until further evidence for their reliability has been provided. That is, the current state of evidence does not support the common claim that they can replace (controlled) experiments and case studies.
2. From a research perspective, future studies should not just focus on replicating reference behavior, as many studies have shown that this was successfully achieved. In particular, future studies should go through all necessary steps (building and calibrating models based on a real-world context, establish structural and behavioral model validity, and conduct a series of evaluations of the simulation with respect to its purpose) to significantly drive forward the field of software process simulation. Here, several studies are needed to cover the different purposes more extensively.
3. The success of step 2) depends very much on complete reporting of how the research study is conducted, in particular with respect to reporting context, data collection, and validity. Important references guiding simulation researchers are Petersen and Wohlin for describing the context in industrial studies [36], Runeson and Höst [41] for case study research, Wohlin et al. [50] on conducting and

reporting experiments. For reporting SPSM studies using continuous simulation a good template is available in Madachy's book [30].

2.7 Conclusion

In this study, we have identified literature reporting the application of software process simulation in real-world settings. We identified a total of 87 primary studies. To increase the validity of the results we undertook a number of measures including individually selecting and assessing the articles, conducting pilots in each phase, calculating inter-rater agreement and discussing any case of disagreement.

Articles were assessed based on their reporting for scientific rigor and industrial relevance. Furthermore, we evaluated whether the simulation model's validity was assured. We also determined how studies evaluated simulation against the purpose for which it was used in the real-world. A large majority of the primary studies scored poorly with respect to the rigor and relevance criteria.

With regard to the scoping of real-world simulation studies we conclude that a research gap was identified in relation to simulation of concurrent projects, and the study of long term evolution from a product and organizational perspective. Such projects are of particular interest in the case of large scale software development where several teams work concurrently in software projects for development of an overall system.

Figure 2.3 provides an overview of achieved validation and evaluation of usefulness. The figure shows that 18% of the primary studies verified the model structure and behavior. Overall, 13% provided an evaluation and only four of these studies scored high on either rigor or relevance criteria. Furthermore, the evaluation done was at best only static according to the definition from [15], i.e. feedback from practitioners was collected whether the model has the potential to fulfil the intended purpose. Of the overall set, only one article reports having verified structure and behavior of the model, and evaluated it against the specified purpose.

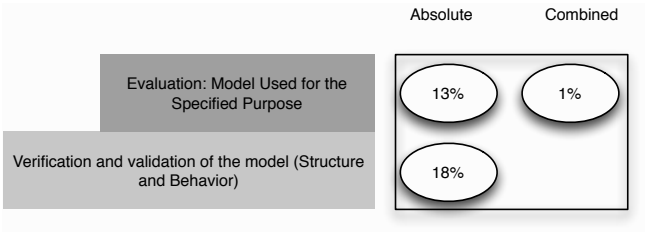


Figure 2.3: State of model validation and evaluation

Based on the results of this systematic literature review that was more extensive in coverage than any of the previously published secondary studies we can draw the following conclusions:

- Despite the large number of industrial applications found in this review, there are no reported cases of the transfer of technology where SPSM was successfully transferred to practitioners in the software industry. Furthermore, there are no studies reporting a long-term use of SPSM in practice. There is no evidence to back the claims of practical adoption and impact on industrial practice [52, 53]. This finding supports the position taken by Pfahl [37] that there is little evidence that process simulation has become an accepted and regularly used tool in industry.
- Based on the reported cost of conducting an SPSM based study (in a few studies), SPSM is not an inexpensive undertaking. Furthermore, without first reporting the cost of adopting SPSM we cannot have any discussion about the cost of *"not simulating"* [8].
- There is no conclusive evidence to substantiate the claimed benefits of SPSM for any of the proposed purposes. It has been sufficiently argued and claimed in proof-of-concept studies that different simulation approaches *"can be"* used to simulate the software process in varying scopes and that these models *"can be"* used for various purposes. However, the need now is to take the field one step further and provide evidence of these claims by evaluating these research proposals in the real-world. Therefore, while more industry relevant empirical research in SPSM [46] should be the direction, the goal should be to evaluate the usefulness of SPSM in practice.

In future work, based on our findings, it is important to evaluate simulation against the purposes (e.g. education and training, prediction), which has not been done so far. Future studies should not focus on evaluating the ability of simulation to reproduce reference behavior, the fact that it is capable to do this is well established in the literature.

References

[1] R. Ahmed, T. Hall, and P. Wernick, "A proposed framework for evaluating software process simulation models," in *Proceedings of the International Workshop on Software Process Simulation and Modeling (ProSim)*, 2003.

[2] R. Ahmed, T. Hall, P. Wernick, and S. Robinson, "Evaluating a rapid simulation modelling process (RSMP) through controlled experiments," in *Proceedings of the International Symposium on Empirical Software Engineering*, Piscataway, NJ, USA, 2005.

[3] R. Ahmed, T. Hall, P. Wernick, S. Robinson, and M. Shah, "Software process simulation modelling: A survey of practice," *Journal of Simulation*, vol. 2, no. 2, pp. 91–102, 2008.

[4] N. b. Ali, K. Petersen, and C. Wohlin, "Supporting information for the systematic literature review," 2014. [Online]. Available: http://www.bth.se/com/nal.nsf/pages/spsm

[5] X. Bai, H. Zhang, and L. Huang, "Empirical research in software process modeling: A systematic literature review." in *Proceedings of the International Symposium on Empirical Software Engineering and Measurement ESEM*, 2011, pp. 339–342.

[6] O. Balci, "Guidelines for successful simluation studies," in *Proceedings of the Winter Simulation Conference*. IEEE Press, 1990, pp. 25–32.

[7] V. Basili, "Quantitative evaluation of software methodology," in *Proceedings of the First Pan Pacific Computer Conference, Melbourne, Australia*, 1985.

[8] T. Birkhölzer, "Software process simulation is simulation too - what can be learned from other domains of simulation?" in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 223–225.

[9] Z. Car and B. Mikac, "A method for modeling and evaluating software maintenance process performances," in *Proceedings of the 6th European Conference on Software Maintenance and Reengineering*, 2002, pp. 15–23.

[10] J. de Almeida Biolchini, P. Mian, A. Natali, T. Conte, and G. Travassos, "Scientific research ontology to support systematic review in software engineering," *Advanced Engineering Informatics*, vol. 21, no. 2, pp. 133–151, 2007.

[11] F. Deissenboeck and M. Pizka, "The economic impact of software process variations," in *Proceedings of the 2007 International Conference on Software Process*. Springer-Verlag, 2007, pp. 259–271.

[12] T. Dybå, "Contextualizing empirical evidence," *IEEE Software*, vol. 30, no. 1, pp. 81–83, 2013.

[13] S. Ferreira, J. Collofello, D. Shunk, G. Mackulak, and P. Wolfe, "Utilization of process modeling and simulation in understanding the effects of requirements volatility in software development," in *Proceedings of the International Workshop on software process Simulation and Modeling (ProSim)*, 2003.

[14] B. B. N. d. França and G. H. Travassos, "Are we prepared for simulation based studies in software engineering yet?" *CLEI Electronic Journal*, vol. 16, no. 1, pp. 9–9, Apr. 2013.

[15] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice," *IEEE Software*, vol. 23, no. 6, pp. 88–95, 2006.

[16] D. Houston, "Research and practice reciprocity in software process simulation," in *Proceedings of the International Conference on Software and System Process (ICSSP)*. IEEE, 2012, pp. 219–220.

[17] ISO/IEC, "ISO/IEC 15504 information technology – process assessment (parts 1–5)," 2003–2006.

[18] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empirical Software Engineering*, vol. 16, no. 3, pp. 365–395, 2010.

[19] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling: Why? what? how?" *Journal of Systems and Software*, vol. 46, pp. 91–105, 1999.

[20] K. S. Khan, G. ter Riet, J. Glanville, A. J. Sowden, and J. Kleijnen, *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews*, 2nd ed. York, United Kingdom: NHS Centre for Reviews and Dissemination, University of York, 2001, vol. 4.

[21] B. Kitchenham, "What's up with software metrics? - a preliminary mapping study," *Journal of Systems and Software*, vol. 83, no. 1, pp. 37–51, Jan. 2010.

[22] B. Kitchenham, P. Brereton, and D. Budgen, "Mapping study completeness and reliability - a case study," in *Proceedings of the 16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, 2012, pp. 126–135.

[23] B. Kitchenham, P. Brereton, Z. Li, D. Budgen, and A. Burn, "Repeatability of systematic literature reviews," in *Proceedings of the 15th Annual Conference on Evaluation Assessment in Software Engineering (EASE)*, 2011, pp. 46–55.

[24] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 2007.

[25] S. Kusumoto, O. Mizuno, T. Kikuno, Y. Hirayama, Y. Takagi, and K. Sakamoto, "A new software project simulator based on generalized stochastic petri-net," in *Proceedings of the 19th International Conference on Software Engineering*. ACM, 1997, pp. 293–302.

[26] A. M. Law, "How to build valid and credible simulation models," in *Proceeding of the 2001 Winter Simulation Conference*. IEEE, 2001, pp. 22–29.

[27] D. Liu, Q. Wang, and J. Xiao, "The role of software process simulation modeling in software risk management: A systematic review," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 302–311.

[28] J. Münch, "Evolving process simulators by using validated learning," in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 226–227.

[29] R. Madachy, "Simulation," in *Encyclopedia of Software Engineering*, J. J. Marciniak, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., Jan. 2002.

[30] R. J. Madachy, *Software Process Dynamics*. Hoboken, NJ, USA: Wiley-IEEE Press, 2008.

[31] ——, "System dynamics modeling of an inspection-based process," in *Proceedings of the 18th International Conference on Software Engineering*. IEEE, 1996, pp. 376–386.

[32] S. Mujtaba, R. Feldt, and K. Petersen, "Waste and lead time reduction in a software product customization process with value stream maps," in *Proceedings*

of the 21st Australian Software Engineering Conference (ASWEC), 2010, pp. 139–148.

[33] S. Park, H. Kim, D. Kang, and D.-H. Bae, "Developing a simulation model using a spem-based process model and analytical models," in *Proceedings of the 4th International Workshop CIAO! and 4th International Workshop EOMAS*. Springer, 2008, pp. 164–178.

[34] S. H. Park, K. S. Choi, K. Yoon, and D.-H. Bae, "Deriving software process simulation model from spem-based software process model," in *Proceedings of the 14th Asia-Pacific Software Engineering Conference*. IEEE, 2007, pp. 382–389.

[35] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 71–80, 2008.

[36] K. Petersen and C. Wohlin, "Context in industrial software engineering research," *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 401–404, Oct. 2009.

[37] D. Pfahl, "Process simulation: A tool for software project managers?" in *Software Project Management in a Changing World*, G. Ruhe and C. Wohlin, Eds. Springer Berlin Heidelberg, 2014, pp. 425–446.

[38] D. Raffo, "Process simulation will soon come of age: Where's the party?" in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 231–231.

[39] D. M. Raffo, R. Ferguson, S.-O. Setamanit, and B. D. Sethanandha, "Evaluating the impact of the QuARS requirements analysis tool using simulation," in *Proceedings of the International Conference on Software Process*, ser. ICSP'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 307–319.

[40] M. Ruiz, I. Ramos, and M. Toro, "A dynamic integrated framework for software process improvement," *Software Quality Journal*, vol. 10, no. 2, pp. 181–194, 2002.

[41] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[42] I. Rus, H. Neu, and J. Münch, "A systematic methodology for developing discrete event simulation models of software development processes," in *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*, 2003.

[43] R. E. Shannon, "Intelligent simulation environments." in *Proceedings of the Conference on Intelligent Simulation Environments*, 1986, pp. 150–156.

[44] F. Shull, J. Singer, and D. Sjø berg, *Guide to advanced empirical software engineering*. Springer, 2007.

[45] D. C. Slawson, "How to read a paper: The basics of evidence based medicine," *BMJ*, vol. 315, no. 7112, p. 891, 1997.

[46] S. Sutton, "Advancing process modeling, simulation, and analytics in practice," in *Proceedings of the International Conference on Software and System Process (ICSSP)*, 2012, pp. 221–222.

[47] C. P. Team, "CMMI for development, version 1.2," Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA, 2006.

[48] I. Turnu, M. Melis, A. Cau, M. Marchesi, and A. Setzu, "Introducing TDD on a free libre open source software project: a simulation experiment," in *Proceedings of the 2004 Workshop on Quantitative Techniques for Software Agile Process*. ACM, 2004, pp. 59–65.

[49] C. von Wangenheim and F. Shull, "To game or not to game?" *IEEE Software*, pp. 92–94, 2009.

[50] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer, 2012.

[51] C. Wohlin, P. Runeson, P. A. da Mota Silveira Neto, E. Engström, I. do Carmo Machado, and E. S. de Almeida, "On the reliability of mapping studies in software engineering," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2594 – 2610, 2013.

[52] H. Zhang, "Special panel: Software process simulation—at a crossroads?" in *Proceedings of the International Conference on Software and System Process (ICSSP)*. IEEE, 2012, pp. 215–216.

[53] H. Zhang, R. Jeffery, D. Houston, L. Huang, and L. Zhu, "Impact of process simulation on software practice: an initial report," in *Proceedings of the 33rd International Conference on Software Engineering (ICSE)*, 2011, pp. 1046–1056.

[54] H. Zhang, R. Jeffery, and L. Zhu, "Hybrid modeling of test-and-fix processes in incremental development," in *Proceedings of the International Conference on Software Process (ICSP)*. Springer, 2008, pp. 333–344.

[55] H. Zhang, B. Kitchenham, and R. Jeffery, "Qualitative vs. quantitative software process simulation modeling: Conversion and comparison," in *Proceedings of the Australian Software Engineering Conference*. IEEE, 2009, pp. 345–354.

[56] H. Zhang, B. Kitchenham, and D. Pfahl, "Software process simulation modeling: an extended systematic review," in *Proceedings of the International Conference on Software Process (ICSP)*. Springer, 2010, pp. 309–320.

[57] ——, *Reflections on 10 years of software process simulation modeling: a systematic review*. Springer Berlin Heidelberg, 2008.

[58] ——, "Software process simulation modeling: Facts, trends and directions," in *Proceedings of the 15th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2008, pp. 59–66.

[59] ——, "Software process simulation over the past decade: trends discovery from a systematic review," in *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Kaiserslautern, Germany: ACM, 2008, pp. 345–347.

Primary studies in the systematic literature review

[60] T. Abdel-Hamid, "The economics of software quality assurance: a simulation-based case study," *MIS Quarterly*, vol. 12, no. 3, pp. 395–411, 1988.

[61] ——, "Understanding the "90% syndrome" in software project management: A simulation-based case study," *Journal of Systems and Software*, vol. 8, no. 4, pp. 319–330, 1988.

[62] ——, "The dynamics of software project staffing: a system dynamics based simulation approach," *IEEE Transactions on Software Engineering*, vol. 15, no. 2, pp. 109–119, 1989.

[63] ——, "On the utility of historical project statistics for cost and schedule estimation: results from a simulation-based case study," *Journal of Systems and Software*, vol. 13, no. 1, pp. 71–82, 1990.

[64] T. Abdel-Hamid and F. Leidy, "An expert simulator for allocating the quality assurance effort in software development," *Simulation*, vol. 56, no. 4, pp. 233–240, 1991.

[65] T. Abdel-Hamid, "A multiproject perspective of single-project dynamics," *Journal of Systems and Software*, vol. 22, no. 3, pp. 151–165, 1993.

[66] T. K. Abdel-Hamid, "A study of staff turnover, acquisition, and assimiliation and their impact on software development cost and schedule," *Journal of Management Information Systems*, vol. 6, no. 1, pp. 21–40, 1989.

[67] A. Al-Emran, P. Kapur, D. Pfahl, and G. Ruhe, "Simulating worst case scenarios and analyzing their combined effect in operational release planning," in *Proceedings of the International Conference on Software Process, ICSP*, vol. 5007. Springer, 2008, p. 269.

[68] ——, "Studying the impact of uncertainty in operational release planning - an integrated method and its initial evaluation," *Information and Software Technology*, vol. 52, no. 4, pp. 446–461, 2010.

[69] D. J. Anderson, G. Concas, M. I. Lunesu, M. Marchesi, and H. Zhang, "A comparative study of scrum and kanban approaches on a real case study using simulation," in *Proceedings of the 13th International Conference on Agile Software Development, XP 2012, May 21, 2012 - May 25, 2012*, vol. 111 LNBIP. Springer Verlag, 2012, Conference Proceedings, pp. 123–137.

[70] C. Andersson, L. Karlsson, J. Nedstam, M. Höst, and B. Nilsson, "Understanding software processes through system dynamics simulation:a case study," in *Proceedings of the 9th Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*. IEEE, 2002, pp. 41–48.

[71] I. Antoniades, I. Stamelos, L. Angelis, and G. Bleris, "A novel simulation model for the development process of open source software projects," *Software Process: Improvement and Practice*, vol. 7, no. 3-4, pp. 173–188, 2002.

[72] G. Antoniol, A. Cimitile, G. Di Lucca, and M. Di Penta, "Assessing staffing needs for a software maintenance project through queuing simulation," *IEEE Transactions on Software Engineering*, vol. 30, no. 1, pp. 43–58, 2004.

[73] E. Aranha and P. Borba, "Using process simulation to assess the test design effort reduction of a model-based testing approach," in *Proceedings of the International Conference on Software process(ICSP)*. Springer-Verlag, 2008, pp. 282–293.

[74] J. Ba and S. Wu, "SdDirM: A dynamic defect prediction model," in *Proceedings of 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA*. IEEE Computer Society, 2012, Conference Proceedings, pp. 252–256.

[75] X. Bai, L. Huang, and S. Koolmanojwong, "Incremental process modeling through stakeholder based hybrid process simulation," in *Proceedings of the International Conference on Software Process, (ICSP)*. Berlin, Germany: Springer-Verlag, 2009, pp. 280–292.

[76] T. Birkhölzer, C. Dickmann, J. Vaupel, and L. Dantas, "An interactive software management simulator based on the CMMI framework," *Software Process: Improvement and Practice*, vol. 10, no. 3, pp. 327–340, 2005.

[77] L. Cao, B. Ramesh, and T. Abdel-Hamid, "Modeling dynamics in agile software development," *ACM Transactions on Management Information Systems*, vol. 1, no. 1, 2010.

[78] Z. Car, B. Mikac, and V. Sinkovic, "A simulation method for telecommunication software maintenance process," in *Proceedings of the 20th IASTED International Conference Applied Informatics AI*, 2002, pp. 69–74.

[79] B. Chatters, M. Lehman, J. Ramil, and P. Wernick, "Modelling a software evolution process: a long-term case study," *Software Process: Improvement and Practice*, vol. 5, no. 2-3, pp. 91 – 102, 2000.

[80] Y. M. Chen and C.-W. Wei, "Multiagent approach to solve project team work allocation problems," *International Journal of Production Research*, vol. 47, no. 13, pp. 3453–3470, 2009.

[81] E. Chiang and T. Menzies, "Simulations for very early lifecycle quality evaluations," *Software Process: Improvement and Practice*, vol. 7, no. 3-4, pp. 141–159, 2002.

[82] A. M. Christie and M. J. Staley, "Organizational and social simulation of a software requirements development process," *Software Process: Improvement and Practice*, vol. 5, no. 2-3, pp. 103–110, 2000.

[83] D. Crespo and M. Ruiz, "Decision making support in CMMI process areas using multiparadigm simulation modeling," in *2012 Winter Simulation Conference (WSC 2012), 9-12 Dec. 2012*, ser. Proceedings of the Winter Simulation Conference (WSC). IEEE, 2012, Conference Proceedings, p. 12 pp.

[84] J. Dasgupta, G. Sahoo, and R. P. Mohanty, "Monte carlo simulation based estimations: Case from a global outsourcing company," in *Proceedings of the 1st International Technology Management Conference (ITMC)*, ser. Proceedings of the 1st International Technology Management Conference, ITMC 2011. IEEE Computer Society, 2011, Conference Proceedings, pp. 619–624.

[85] F. Deissenboeck and M. Pizka, "Probabilistic analysis of process economics," *Software Process: Improvement and Practice*, vol. 13, no. 1, pp. 5–17, 2008.

[86] M. Di Penta, M. Harman, and G. Antoniol, "The use of search-based optimization techniques to schedule and staff software projects: An approach and an empirical study," *Software - Practice and Experience*, vol. 41, no. 5, pp. 495–519, 2011.

[87] C. Dickmann, H. Klein, T. Birkhölzer, W. Fietz, J. Vaupel, and L. Meyer, "Deriving a valid process simulation from real world experiences," in *Proceedings of the International Conference on Software Process*. Springer, 2007, pp. 272–282.

[88] M. Farshchi, Y. Y. Jusoh, and M. A. A. Murad, "Impact of personnel factors on the recovery of delayed software projects: A system dynamics approach," *Computer Science and Information Systems*, vol. 9, no. 2, pp. 627–651, 2012.

[89] S. Ferreira, J. Collofello, D. Shunk, and G. Mackulak, "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation," *Journal of Systems and Software*, vol. 82, no. 10, pp. 1568–1577, 2009.

[90] J. Ghosh, "Concurrent, overlapping development and the dynamic system analysis of a software project," *IEEE Transactions on Engineering Management*, vol. 57, no. 2, pp. 270–287, 2010.

[91] M. L. Gillenson, M. J. Racer, S. M. Richardson, and X. Zhang, "Engaging testers early and throughout the software development process: Six models and a simulation study," *Journal of Information Technology Management*, vol. 22, no. 1, p. 8Äì27, 2011.

[92] M. Höst, B. Regnell, J. Natt och Dag, J. Nedstam, and C. Nyberg, "Exploring bottlenecks in market-driven requirements management processes with discrete event simulation," *Journal of Systems and Software*, vol. 59, no. 3, pp. 323–332, 2001.

[93] D. Houston, "An experience in facilitating process improvement with an integration problem reporting process simulation," *Software Process: Improvement and Practice*, vol. 11, no. 4, pp. 361–371, 2006.

[94] D. X. Houston, G. T. Mackulak, and J. S. Collofello, "Stochastic simulation of risk factor potential effects for software development risk management," *Journal of Systems and Software*, vol. 59, no. 3, pp. 247–257, 2001.

[95] N. Hsueh, W. Shen, Z. Yang, and D. Yang, "Applying UML and software simulation for process definition, verification, and validation," *Information and Software Technology*, vol. 50, no. 9, pp. 897–911, 2008.

[96] L. Huang, B. Boehm, H. Hu, J. Ge, J. Lü, and C. Qian, "Applying the value/petri process to ERP software development in china," in *Proceedings of the 28th International Conference on Software Engineering*. ACM, 2006, pp. 502–511.

[97] H. Jian-Hong, B. Xiaoing, L. Qi, and X. Daode, "A fuzzy-ECM approach to estimate software project schedule under uncertainties," in *Proceedings of the 9th IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW)*, 2011, pp. 316–321.

[98] W. Junjie, L. Juan, W. Qing, Z. He, and W. Haitao, "A simulation approach for impact analysis of requirement volatility considering dependency change," in *Proceedings of the 18th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ*. Springer-Verlag, 2012, Conference Proceedings, pp. 59–76.

[99] E. Katsamakas and N. Georgantzas, "Why most open source development projects do not succeed?" in *Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development*. IEEE, 2007, pp. 3–3.

[100] S. Kusumoto, O. Mizuno, T. Kikuno, Y. Hirayama, Y. Takagi, and K. Sakamoto, "Software project simulator for effective process improvement," *Journal of Information Processing Society of Japan*, vol. 42, no. 3, pp. 396–408, 2001.

[101] X. M. Li, Y. L. Wang, L. Y. Sun, and L. Li, "Modeling uncertainties involved with software development with a stochastic petri net," *Expert Systems*, vol. 23, no. 5, pp. 302–312, 2006.

[102] C. T. Lin, "Analyzing the effect of imperfect debugging on software fault detection and correction processes via a simulation framework," *Mathematical and Computer Modelling*, vol. 54, no. 11-12, pp. 3046–3064, 2011.

[103] C. Lin and R. Levary, "Computer-aided software development process design," *IEEE Transactions on Software Engineering*, vol. 15, no. 9, pp. 1025–1037, 1989.

[104] C. Y. Lin, T. Abdel-Hamid, and J. S. Sherif, "Software-engineering process simulation model (SEPS)," *Journal of Systems and Software*, vol. 38, no. 3, pp. 263–277, 1997.

[105] R. J. Madachy, "Knowledge-based risk assessment and cost estimation," *Automated Software Engineering*, vol. 2, no. 3, pp. 219–230, 1995.

[106] R. J. Madachy and B. Khoshnevis, "Dynamic simulation modeling of an inspection-based software lifecycle process," *Simulation*, vol. 69, no. 1, pp. 35–47, 1997.

[107] R. Martin and D. Raffo, "Application of a hybrid process simulation model to a software development project," *Journal of Systems and Software*, vol. 59, no. 3, pp. 237–246, 2001.

[108] J. A. McCall, G. Wong, and A. Stone, "A simulation modeling approach to understanding the software development process." in *Proceedings of the 4th Annual Software Engineering Workshop, Goddard Space Flight Center Greenbelt, Maryland*, 1979, pp. 250–273.

[109] X. Meilong, L. Congdong, and C. Jie, "System dynamics simulation to support decision making in software development project," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM*, 2008, pp. 1–4.

[110] M. Melis, I. Turnu, A. Cau, and G. Concas, "Evaluating the impact of test-first programming and pair programming through software process simulation," *Software Process: Improvement and Practice*, vol. 11, no. 4, pp. 345–360, 2006.

[111] C. Mizell and L. Malone, "A software development simulation model of a spiral process," *International Journal of Software Engineering*, vol. 2, no. 2, 2007.

[112] ——, "A project management approach to using simulation for cost estimation on large, complex software development projects," *Engineering Management Journal*, vol. 19, no. 4, pp. 28 – 34, 2007.

[113] O. Mizuno, S. Kusumoto, T. Kikuno, Y. Takagi, and K. Sakamoto, "Estimating the number of faults using simulator based on generalized stochastic petri-net model," in *Proceedings of the 6th Asian Test Symposium (ATS'97)*. IEEE, 1997, pp. 269–274.

[114] O. Mizuno, D. Shimoda, T. Kikuno, and Y. Takagi, "Enhancing software project simulator toward risk prediction with cost estimation capability," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 84, no. 11, pp. 2812–2821, 2001.

[115] M. Nakatani and S. Nishida, "Trouble communication model in a software development project," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 75, no. 2, pp. 196–206, 1992.

[116] E. Paikari, G. Ruhe, and P. H. Southekel, "Simulation-based decision support for bringing a project back on track: The case of rup-based software construction," in *Proceedings of the International Conference on Software and System Process, ICSSP*, 2012, Conference Proceedings, pp. 13–22.

[117] S. Park, H. Kim, D. Kang, and D. Bae, "Developing a software process simulation model using SPEM and analytical models," *International Journal of Simulation and Process Modelling*, vol. 4, no. 3, pp. 223–236, 2008.

[118] D. Pfahl and K. Lebsanft, "Knowledge acquisition and process guidance for building system dynamics simulation models: an experience report from software industry," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 04, pp. 487–510, 2000.

[119] ——, "Using simulation to analyse the impact of software requirement volatility on project performance," *Information and Software Technology*, vol. 42, no. 14, pp. 1001–1008, 2000.

[120] D. Pfahl, M. Stupperich, and T. Krivobokova, "PL-SIM: A generic simulation model for studying strategic spi in the automotive industry," in *Proceedings of the International Workshop on Software Process Simulation and Modeling (ProSim)*, 2004, pp. 149–158.

[121] I. Podnar and B. Mikac, "Software maintenance process analysis using discrete-event simulation," in *Proceedings of the Fifth European Conference on Software Maintenance and Reengineering*. IEEE, 2001, pp. 192–195.

[122] J. E. Psaroudakis and A. Eberhardt, "A discrete event simulation model to evaluate changes to a software project delivery process," in *Proceedings of the 13th IEEE Conference on Commerce and Enterprise Computing*. IEEE Computer Society, 2011, Conference Proceedings, pp. 113–20.

[123] D. Raffo, R. Ferguson, S. Setamanit, and B. Sethanandha, "Evaluating the impact of requirements analysis tools using simulation," *Software Process: Improvement and Practice*, vol. 13, no. 1, pp. 63–73, 2008.

[124] D. Raffo, J. Vandeville, and R. Martin, "Software process simulation to achieve higher CMM levels," *Journal of Systems and Software*, vol. 46, no. 2, pp. 163–172, 1999.

[125] D. M. Raffo, U. Nayak, S.-o. Setamanit, P. Sullivan, and W. Wakeland, "Using software process simulation to assess the impact of IV&V activities," in *Proceedings of the 5th International Workshop on Software Process Simulation and Modeling (ProSim)*, 2004, pp. 197–205.

[126] D. M. Raffo, "Software project management using PROMPT: A hybrid metrics, modeling and utility framework," *Information & Software Technology*, vol. 47, no. 15, pp. 1009–1017, 2005.

[127] H. Rahmandad and D. M. Weiss, "Dynamics of concurrent software development," *System Dynamics Review (Wiley)*, vol. 25, no. 3, pp. 224–249, 2009.

[128] J. F. Ramil and N. Smith, "Qualitative simulation of models of software evolution," *Software Process: Improvement and Practice*, vol. 7, no. 3-4, pp. 95–112, 2002.

[129] B. Regnell, B. Ljungquist, T. Thelin, and L. Karlsson, "Investigation of requirements selection quality in market-driven software processes using an open source discrete event simulation framework," in *Proceedings of the 5th International Workshop on Software Process Simulation and Modeling (ProSim)*, Stevenage, UK, 2004, pp. 84 – 93.

[130] M. Ruiz, I. Ramos, and M. Toro, "A simplified model of software project dynamics," *Journal of Systems and Software*, vol. 59, no. 3, pp. 299–309, 2001.

[131] ——, "Using dynamic modeling and simulation to improve the cots software process," in *Proceedings of the 5th International Conference on Product Focused Software Process Improvement*. Springer, 2004, pp. 568–581.

[132] I. Rus, F. Shull, and P. Donzelli, "Decision support for using software inspections," in *Proceedings of the 28th Annual NASA GoddardSoftware Engineering Workshop*. IEEE, 2003, pp. 3–11.

[133] S.-O. Setamanit and D. Raffo, "Identifying key success factors for globally distributed software development project using simulation: a case study," in *Proceedings of the International Conference on Software process*. Springer-Verlag, 2008, pp. 320–332.

[134] P. Seunghun and B. Doo-Hwan, "An approach to analyzing the software process change impact using process slicing and simulation," *Journal of Systems and Software*, vol. 84, no. 4, pp. 528–43, 2011.

[135] J. Shen, S. Changchien, and T. Lin, "A petri-net based modeling approach to concurrent software engineering tasks." *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 767–795, 2005.

[136] B. Spasic and B. S. S. Onggo, "Agent-based simulation of the software development process: a case study at AVL," in *Proceedings of the Winter Simulation Conference (WSC)*, ser. Proceedings of the 2012 Winter Simulation Conference (WSC 2012). IEEE, 2012, Conference Proceedings, p. 11 pp.

[137] F. Stallinger, "Software process simulation to support ISO/IEC 15504 based software process improvement," *Software Process: Improvement and Practice*, vol. 5, no. 2-3, pp. 197–209, 2000.

[138] R. Tausworthe and M. Lyu, "A generalized software reliability process simulation technique and tool," in *Proceedings of the 5th International Symposium on Software Reliability Engineering, 1994.* IEEE, 1994, pp. 264–273.

[139] I. Turnu, M. Melis, A. Cau, A. Setzu, G. Concas, and K. Mannaro, "Modeling and simulation of open source development using an agile practice," *Journal of Systems Architecture*, vol. 52, no. 11, pp. 610–618, 2006.

[140] Y. Wang and Y. Chen, "An experience of modeling and simulation in support of CMMI process," in *Proceedings of the 15th European Simulation Symposium*, 2003, pp. 297–302.

[141] P. Wernick and M. Lehman, "Software process white box modelling for FEAST/1," *Journal of Systems and Software*, vol. 46, no. 2, pp. 193–201, 1999.

[142] R. T. Wiegner and S. Y. Nof, "The software product feedback flow model for development planning," *Information and Software Technology*, vol. 35, no. 8, pp. 427–438, 1993.

[143] D. Wu, H. Song, M. Li, C. Cai, and J. Li, "Modeling risk factors dependence using copula method for assessing software schedule risk," in *Proceedings of the 2nd International Conference on Software Engineering and Data Mining (SEDM)*. IEEE, 2010, pp. 571–574.

[144] Y. Yong and B. Zhou, "Software process deviation threshold analysis by system dynamics," in *Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME)*. IEEE, 2010, pp. 121–125.

[145] H. Zhang, R. Jeffery, and L. Zhu, "Investigating test-and-fix processes of incremental development using hybrid process simulation," in *Proceedings of the 6th International Workshop on Software Quality*. ACM, 2008, pp. 23–28.

[146] H. Zhang, "Investigating the gap between quantitative and qualitative/semi-quantitative software process simulation models: An explorative study," in *Proceedings of the International Conference on Software Process, (ICSP)*, vol. 5543. Springer, 2009, p. 198.

Chapter 3

Use and evaluation of simulation for software process education: a case study

Abstract

Software Engineering is an applied discipline and its concepts are difficult to grasp only at a theoretical level alone. In the context of a project management course, we introduced and evaluated the use of software process simulation modeling (SPSM) based games for improving students' understanding of software development processes. The effects of the intervention were measured by evaluating the students' arguments for choosing a particular development process. The arguments were assessed with the Evidence-Based Reasoning framework, which was extended to assess the strength of an argument. The results indicate that students generally have difficulty providing strong arguments for their choice of process models. Nevertheless, the assessment indicates that the intervention of the SPSM game had a positive impact on the students' arguments. Even though the illustrated argument assessment approach can be used to provide formative feedback to students, its use is rather costly and cannot be considered a replacement for traditional assessments.

3.1 Introduction

The Software Engineering (SE) discipline spans from technical aspects, such as developing techniques for automated software testing, over defining new processes for software development improvement, to people-related and organizational aspects, such as team management and leadership. This is evident in the software development process, which is *"the coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product"* [10]. This breadth of topics encompassed here makes education in SE challenging as the interaction of the different disciplines cannot be exclusively taught on a theoretical level, but must also be experienced in practice. As such, SE education needs to identify means to prepare students better for their tasks in industry [15].

However, the complexity and dynamism of software processes makes it difficult to illustrate the implications of the chosen development process on the outcomes of a project. Students will have to undertake multiple iterations of developing the same project using different software development processes to understand the various processes and their implication on the project attributes [25]. Such repetitions are however impractical because of the time and cost involved. To overcome this shortcoming software process simulation modeling (SPSM) has been proposed as a means of SE education. SPSM is the numerical evaluation of a computerized-mathematical model that imitates the real-world software development process behavior [12]. It has been found to be useful in SE education as a complement to other teaching methods e.g. in combination with lectures, lab sessions and projects [16, 25].

In this study we motivate, illustrate and evaluate how a targeted change was introduced in the graduate-level *Applied Software Project Management (ASPM)* course. The course aims to convey to students in a hands-on manner how to prepare, execute and finalize a software project. In previous instances of the course, we have observed that students encounter difficulties in choosing an appropriate software development process and in motivating their choice. We hypothesize that the students lack experience of different software development processes, and lack therefore the analytical insight required to choose a process appropriate for the characteristics of the course project. We study our hypothesis by exposing students to software process simulations and by evaluating thereafter the argumentative strength for choosing/discarding a particular process.

There are three major contributions of this study. First, a review of frameworks for evaluating argumentative reasoning was updated to cover more recent research. Secondly the framework relevant for evaluating arguments in the context of SE was selected and adapted. Thirdly, independent of the creators of SimSE, we used it in

the context of an active course instead of a purely experimental setting, and evaluated its effect indirectly, in terms of students' understanding of software development processes.

The remainder of the chapter is structured as follows: Section 3.2 summarizes the relevant work on the topic of SPSM in SE education. Section 3.3 presents the context of the study, research questions, data collection and analysis methods. Section 3.4 presents the results, Section 3.5 revisits the research questions based on the findings and Section 3.6 concludes the chapter.

3.2 Background and related work

In this section, we briefly discuss the two overarching themes in this study: SPSM based education and evaluation on scientific argumentation.

3.2.1 SPSM in SE education

SPSM provides an alternative to manipulation of the actual software process by providing a test-bed for experimentation with realistic considerations. Compared to static and analytical models, SPSM achieves this because of its ability to capture the underlying complexity in software development by representing uncertainty, dynamic behavior and feedback/feed-forward mechanisms [12].

Since the initial proposal of SPSM its potential as a means of education and training was recognized [12]. Some of the claimed benefits of SPSM for SE education include: increased interest in SE project management [17], motivation of students [7], and effective learning [19]. It can facilitate understanding by experiencing different processes with certain roles (e.g. as a as a software manager making decisions in software development, which would not have been possible in an academic context without SPSM [26]).

Navarro and Hoek [16] evaluated the experience of students playing SPSM based games for SE education. They found that the SPSM based teaching is applicable for various types of learners as it aligns well with objectives of a multitude of learning theories. For example, it encourages exploratory learning by experimenting, emphasizes learning by doing and through failure, and by embedding in a context that resembles the real-world use of the phenomenon of interest.

Wangenheim and Shull [25], in a systematic literature review of studies using SPSM for SE education, found that the two most frequent aims in such studies are *"SE Project Management"* and *"SE process"* knowledge [25]. They also found that in most of the existing research, subjective feedback was collected after the students

had used the game [25]. Similarly, they reported that it was difficult to evaluate the effectiveness of SPSM interventions because a majority of the articles do not report the *"expected learning outcome and the environment in which students used the game"* [25].

These findings motivated our choice to have a simulation based intervention in the course as the two major learning objectives for the course are related to project and process management. The context is described in Section 3.3.1. Furthermore, adhering to the recommendation that is based on empirical studies [25], we used SPSM to target a *"specific learning need"* of the students, i.e. to improve the understanding and implications of a software development life-cycle process. SimSE was the chosen platform due to a stable release, good graphical user-interface and good feedback from earlier evaluations [16]. Unlike the existing evaluations of SimSE, in this study, we took an indirect approach to see if the simulation based intervention had the desired impact. We looked at the quality of arguments for the choice of the life-cycle process in the student reports without explicitly asking them to reflect on the SPSM game.

3.2.2 Evaluating scientific argumentation

Argumentation is a fundamental driver of the scientific discourse, through which theories are constructed, justified, challenged and refuted [9]. However, scientific argumentation has also cognitive values in education, as the process of externalizing one's thinking fosters the development of knowledge [9]. As students mature and develop competence in a subject, they pass through the levels of understanding described in the SOLO taxonomy [4]. In the taxonomy's hierarchy, the quantitative phase (unistructural and multistructural levels) is where students increase their knowledge, whereas in the qualitative phase (relational and extended abstract levels) students deepen their knowledge [3]. The quality of scientific argumentation, which comprises skills residing in higher levels of the SOLO taxonomy, is therefore a reflection of the degree of understanding and competence in a subject.

As argumentation capability and subject competence are intrinsically related, it is important to find means by which scientific argumentation in the context of education can be evaluated. Sampson and Clark [20] provide a review of frameworks developed for the purpose of assessing the nature and quality of arguments. They analyze the studied frameworks along three dimensions of an argument [20]:

1. Structure (i.e., the components of an argument)
2. Content (i.e., the accuracy/adequacy of an arguments components when evaluated from a scientific perspective)
3. Justification (i.e., how ideas/claims are supported/validated within an argument)

We used the same criteria to update their review with newer frameworks for argument evaluation. This analysis was used to select the framework appropriate for use in this study.

3.3 Research design

3.3.1 Context

The objective of the Applied Software Project Management (ASPM) course is to provide students with an opportunity to apply and sharpen their project management skills in a sheltered but still realistic environment. Students participating in ASPM typically[1] have completed a theory-building course on software project management, which includes an introduction to product management, practical project management guided by the Project Management Body of Knowledge [28], and an excursion to leadership in project teams [11].

Figure 3.1 shows the student characteristics of the two course instances that were studied. In 2012, without SPSM intervention, 16 students participated in total, having accumulated on average 18 ECTS points at the start of the course. In 2013, with the SPSM intervention, 15 students participated in total, having accumulated on average 84 ECTS points at the start of the course. In both course instances, three students did not take the theory course on software project management (Advanced SPM). The major difference between the two student groups is that in 2013, considerably more students did not successfully complete the Advanced SPM course. The higher ECTS average in 2013 can be explained by the participation of three Civil Engineering students who chose Applied SPM at the end of their study career while SE and Computer Science students chose the course early in their studies.

The course follows the three months schedule shown in Figure 3.2, which illustrates also the planned interactions between students and instructors. The introduced modifications are shown in italics and further discussed in Section 3.3.3. Students are expected to work 200 hours for this course, corresponding to a 20 hours/week commitment.

The course has five assignments but Assignment 1 and 5 are important for this study (see Figure 3.2). Assignment 1 consists of delivering a project management plan (PMP) where students also report the choice and rationale for a software process they will use. The teams receive oral feedback and clarifications on the PMP during the same week. The course concludes with a presentation where project teams demo their

[1] ASPM is also an optional course in the curriculum for students from computer science and civil engineering programs

Program	2012	2013
Master in Software Engineering	12	12
Master in Computer Science	4	0
Civil Engineer in Industrial Economy	0	3
ECTS (average)	18	84

Advanced SPM grade	2012	2013
A	0	0
B	2	0
C	5	3
D	2	3
E	3	1
Not finished	1	5
Not taken	3	3

Figure 3.1: Student demographics from 2012 (without intervention) and 2013 (with SPSM intervention) of the Applied SPM course

developed products. In Assignment 5, the students are asked to individually answer a set of questions that, among other aspects, inquiry their experience with the used software process in the project.

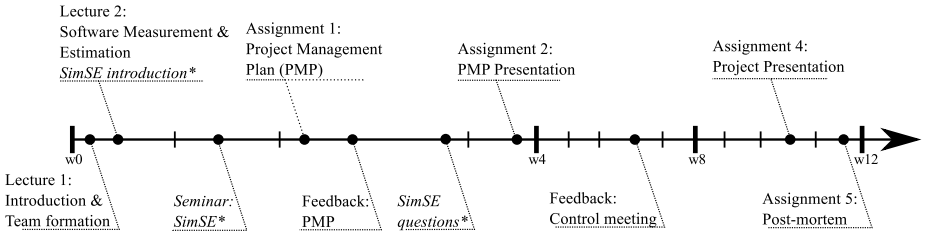
3.3.2 Research questions

The posed research questions in this study are:

RQ1: How can improvement in software development process understanding be assessed?

RQ2: To what extent can process simulation improve students' understanding of software development processes?

With RQ2, we investigate whether process simulation has a positive impact on students' understanding of development processes. Even though studies with a similar



* The SimSE introduction, seminar and questions are the newly introduced treatments

Figure 3.2: ASPM course time-line with events

aim have already been conducted (c.f. [17]), experiments in general are prone to the Hawthorne effect [6], where subjects under study modify their behavior knowing that they are being observed. Similar limitations can be observed in earlier evaluations of SimSE where *"the students were given the assignment to play three SimSE models and answer a set of questions concerning the concepts the models are designed to teach"* [16]. Hence we introduce process simulation as an additional teaching and learning activity into a course whose main purpose is *not* to teach software development processes. Furthermore, we do not modify the requirements for the graded deliverables. Formally, we stated the following hypotheses:

H_0: There is no difference in the students' understanding of process models in course instances 2012 and 2013.

H_a: There is a difference in the students' understanding of process models in course instances 2012 and 2013.

Due to the subtle modifications in the course, we needed to identify new means to evaluate the intervention, measuring the impact of introducing process simulation on students' understanding of development processes. In order to answer RQ1, we update the review by Sampson and Clark [20] with two more recent frameworks proposed by Reznitskaya et al. [18] and Brown et al. [5], select the framework that provides the strongest argument evaluation capabilities, and adapt it to the software engineering context.

In order to answer RQ2, we apply the chosen argument evaluation framework on artifacts delivered by project teams and individual students which did receive the treatments shown in Figure 3.2 and on artifacts delivered in the previous year.

3.3.3 Instrumentation and data collection

Assignment 5: Post mortem, as shown in Figure 3.2, is an individual assignment where students had to motivate and reflect on their choice of software process model selected in their projects. This assignment is used to evaluate the influence of SimSE on the student's understanding of the software processes. A baseline for typical process understanding of students from the course was established by evaluating *Assignment 5* from year 2012 and it was compared to the evaluation results of *Assignment 5* from year 2013. To supplement the analysis we also used *Assignment 1: Project Management Plan (PMP)* (which is a group assignment) from both years. The design for the study is shown in Figure 3.3. Where deltas *'a'* and *'b'* are changes in understanding between the Assignments 1 and 5 within a year. While deltas *'c'* and *'d'* represent changes across the years for Assignment 1 and 5 respectively.

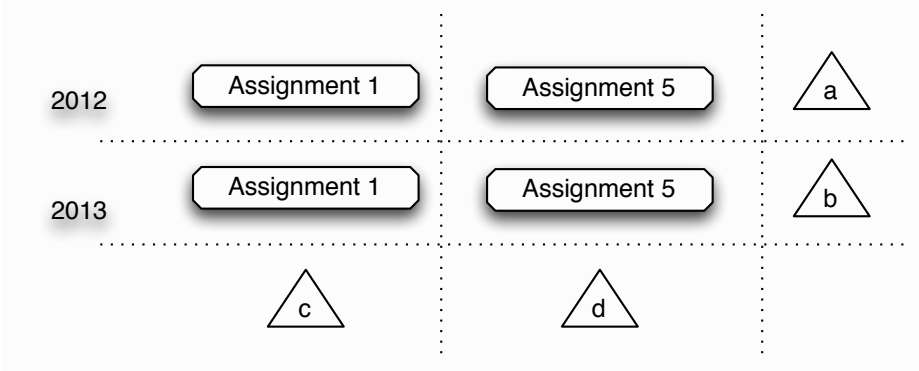


Figure 3.3: Design to evaluate the impact of the SPSM intervention

For the evaluation of assignments, we used the EBR framework [5]. Other frameworks considered and the reasons for this choice are summarised in Section 3.4.2. Once the framework had been adapted, first it was applied on one assignment using *"Think-aloud protocol"* where the authors expressed their thought process while applying the evaluation instrument. This helped to identify ambiguities in the instrument and also helped to develop a shared understanding of it. A pilot of the instrument was done on two assignments where the authors applied it separately and then compared and discussed the results. Both authors individually coded all the assignments and then the codes were consolidated with consensus. The results of this process are presented in Section 3.4.3.

3.3.4 Limitations

The assignments were retrieved from the Learning Management System and personal identification of students was replaced with a unique identifier to hide their identity from the authors. This was done to avoid any personal bias that the authors may have towards the students as their teachers, in this and other courses. Furthermore, to ensure an unbiased assessment both the overall grades of students and their grades in the assignments were hidden from the authors when the assessment instrument was applied in this study.

To avoid any bias introduced by asking questions directly about the intervention of process simulation, and to have a relative baseline for assignments from 2012, we did not change the assignment descriptors for the year 2013. Thus we tried to measure the

effect of the intervention indirectly by observing the quality of argumentation without explicitly asking students to reflect on the experience from simulation based games.

The intervention was applied in a post-graduate course, rendering experiment-like control of settings and variables impossible. Among other factors, any difference in results could purely be because of the different set of students taking the course in the years 2012 and 2013. However, as discussed in Section 3.3.1 the groups of students were fairly similar thus the results are comparable. Small number of students is also a limitation of this study especially to draw any statistical inference with high confidence.

Similarly, by having the students fill out questions about the various simulation based games we tried to ensure that students have indeed played the games. However, we have no way of ensuring that the students did indeed play the games individually and not share the answers with each other. This limitation could weaken the observable effect of the intervention.

3.4 Results

In this section we report the two main results of our study. In Section 3.4.1 we review two argument evaluation approaches and classify them according to the framework presented in Section 3.2.2. Then we choose one argument evaluation approach and adapt it to our goals (Section 3.4.2), and apply it to students arguments on choosing a particular process model for their project (Section 3.4.3). The data (argument coding and quantification) is available in the supplementary material to this chapter, available at http://www.bth.se/com/mun.nsf/pages/simeval.

3.4.1 Review update

In Table 3.1 we summarize Sampson and Clark's [20] analysis w.r.t. the support various frameworks provide to assess structure, content and justification of an argument. In the rest of the section we report the classification of two newer frameworks as an extension to their review.

Brown et al. [5] propose with the Evidence-based Reasoning (EBR) framework an approach to evaluate scientific reasoning that combines Toulmin's argumentation pattern [24] with Duschl's framework of scientific inquiry [8]. This combination proposes scientific reasoning as a two-step process in which a scientific approach to gather and interpret data results in rules that are applied within a general framework of argumentation [5].

Figure 3.4a shows the structure of the framework, consisting of components that should be present in a strong scientific argument. The strength of an argument can

Table 3.1: Strengths and weaknesses of argument assessment frameworks

Framework	Structure	Content	Justification
Domain-general			
Toulmin [24]	strong	weak	weak
Schwarz et al. [22]	strong	moderate	moderate
Domain-specific			
Zohar and Nemet [27]	weak	moderate	strong
Kelly and Takao [13]	strong	weak	strong
Lawson [14]	strong	weak	strong
Sandoval [21]	weak	strong	strong

be characterized by the components present in the argument. For example, in an unsupported claim (Figure 3.4b), there are no rules, evidences or data that objectively support the claim. An analogy (Figure 3.4c) limits the argumentation on supporting a claim only with instances of data, without analyzing and interpreting the data. In an overgeneralization (Figure 3.4d), analysis and formation of a body of evidence is omitted and data is interpreted directly to formulate rules (theories, relationships) that are not supported by evidence.

The EBR was not designed for a specific scientific context [5] and we classify it therefore as a domain-general framework. It provides strong support for evaluating arguments along structure (i.e. the components of an argument) and justification (i.e. how claims are supported within an argument) dimensions. However, solely identifying rules and evidences components in an argument does not provide an assessment of the

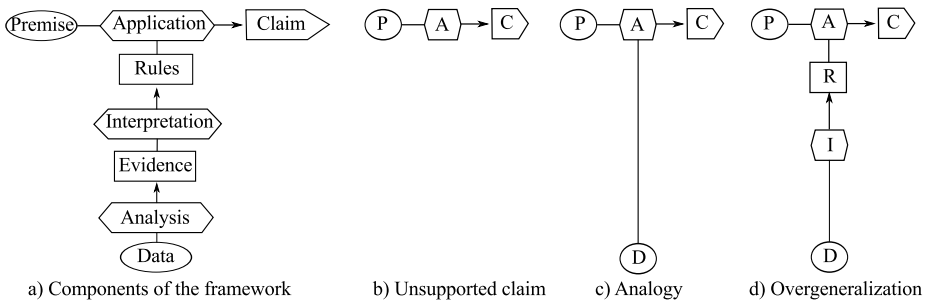


Figure 3.4: The Evidence-Based Reasoning framework (a) and different degrees of argument sophistication (b-d) (adapted from Brown et al. [5])

arguments content, i.e. the adequacy of argument components. As such, the framework provides the tools to identify components of content (rules and evidences), but no direct means to assess the content's quality. Therefore, we rate the framework's support for the content dimension as moderate.

Reznitskaya et al. [18] propose a domain-specific framework for evaluation of written argumentation. They proposed and compared two methods to implement this framework:

1. Analytical method, which is a data driven approach where individual statements are coded and their relevance to the main topic is judged, categories are derived from these codes (deciding about these categories will be based on the theoretical and practical significance in the domain). Next the report is evaluated on five subscales which cover aspects from: number of arguments made, types of previously identified categories of arguments covered in the report, opposing perspectives considered, number of irrelevant arguments and the use of formal aspects of discourse.
2. Holistic method takes a rubric based approach attempting to provide a macro level assessment of the arguments.

In essence, the framework has no explicit focus on the components of an argument and only indirectly covers the aspects of structure while creating the instrument. The fundamental building block of the evaluation framework is the analytical coding process where both the content and justification are considered. Content (accuracy and adequacy) is only assessed by identifying the relevance of the argument to the topic. Justification is covered indirectly in the variety of argument categories identified in the reports. However, all argument categories are given equal weight in scoring. Therefore, we rate the framework support for the structure as weak, and for content and justification as moderate.

3.4.2 Selection and adaptation of an argument evaluation framework

Based on the analysis of the reviewed frameworks, summarized in Table 3.1, and the updated review presented in Section 3.4.1, we decided to use the EBR framework [5]. We chose a domain-general over a domain-specific framework due to our preference of customizing generic principles to a specific context. The alternative, to construct an assessment instrument completely inductively from the particular domain and data, as for example in Reznitskaya et al. [18], would imply a relative assessment approach, weakening the evaluation of the intervention. Furthermore, a domain-generic approach

allows us to re-use the assessment instrument with minor adaptations, lowering the application cost by keeping the general approach intact.

Table 3.2 shows an example argument with the EBR components one can identify in written statements. This example illustrates what we would expect in a strong argument: a general rule on a process model is applied on a premise that refers to the specific circumstances of the students' project, justifying their claim that the selected model was the best choice. The rule is supported by evidence (a reference to a scientific study) and by an experience from the project that creates a relationship between short development cycles and late changes. The evidence is supported by data from the project.

The EBR framework enables a fine-grained deconstruction of arguments into components. The price for this strength on the structural and justification dimension is a moderate support for assessing content (see Section 3.4.1). Even though the overall argument content can be judged to some extent by the interplay between the argument components, domain-specific knowledge is nevertheless required to judge whether individual statements are accurate. Looking at Table 3.2, the rule component in particular requires knowledge on XP in order to decide whether the statement is accurate or not. Hence we assess the argument content by qualifying a rule as sound/unsound, given the stated premise, claim, evidence and data, based on our domain knowledge on process models. Concretely, we declare an argument for a claim as:

- Sound
  - If the stated rule is backed by evidence/data *and* is pertinent for the premise (strong).
  - In arguments with no corroborative evidence (weak): If the stated rule is in compliance with literature and/or the assessors understanding of the topic *and* is pertinent for the premise.

Table 3.2: Example application of the EBR on an ideal argument

Component	Statement
Premise	The requirements for our product are not that clear and likely to change.
Claim	eXtreme Programming (XP) was the best choice for our project.
Rule	XP embraces continuous change by having multiple short development cycles.
Evidence	Reference to Beck [2]; customer changed user interaction requirements six weeks before the delivery deadline but we still delivered a working base product;
Data	Seven change requests to initially stated requirements; four requirements were dropped since customer was satisfied already;

- Unsound
  - If an argument does not fulfill either of the above two criteria.

3.4.3 Application of the chosen framework

In this section we illustrate the results of applying the EBR framework on the students' arguments for choosing/rejecting a particular process model for their project. We coded statements according to the EBR frameworks' components of an argument: a premise (P), rule (R), evidence (E), data (D). We also noted when components are used atomically or are combined into pairs or triples of components to form a coherent argument. Based on this codification, we evaluated the overall content of the argument (unsound / weak / strong) by following the rules established in Section 3.4.2. The claim of the argument, in principle constant and only changing in sign, was that a particular process model is / is not the best choice for the project.

Table 3.3 shows the results in terms of the argument component frequencies encountered in the students' project plans from 2012 (without intervention) and 2013 (with the SPSM intervention). For example, in Plan #1 we identified 8 premises (P), 4 premise-rule (PR) pairs and 1 premise-evidence (PE) pair. We expected to find some premise-rule-evidence (PRE) triples as they would indicate that students can motivate their choice by examples, e.g. by referring to scientific literature, to experience from previous projects or from playing SimSE. However, the results clearly indicate a tendency for students to create overgeneralizing arguments (premise-rule pairs). Looking at the argument content, we identified no strong arguments (lack of evidence component) and, in proportion to weak arguments, a rather large number of unsound arguments, indicating a lack of understanding of process models and their properties.

Table 3.3: Frequencies of identified argument components and argument content strength in project plans for choosing a particular process model

Year	Plan#	P	PR	PE	PRE	RE	R	Unsound	Weak	Strong
	1	8	4	1	0	0	0	2	2	0
2013	2	9	7	0	0	0	3	4	6	0
	3	3	1	0	0	1	0	1	1	0
	Sum	**20**	**12**	**1**	**0**	**1**	**3**	**7**	**9**	**0**
	4	10	3	0	0	0	1	1	3	0
	5	5	7	0	0	1	1	3	6	0
2012	6	6	3	0	0	0	0	0	4	0
	7	2	1	0	0	0	2	1	1	0
	Sum	**23**	**14**	**0**	**0**	**1**	**4**	**5**	**14**	**0**

After assessing the project plans, which were a group assignment, we applied the EBR framework on the project post-mortems that were handed in individually by the students (13 in 2013 and 12 in 2012). Table 3.4 illustrates the results, showing the frequencies of identified argument components and component combinations. Observe that, in contrast to the post-mortem, we identified more combinations of components, e.g. premise-data (PD) or premise-evidence-data (PED) triples. For both years it is evident that students reported more justification components (evidence and data) in the post-mortem than in the project plan. This is expected as we explicitly asked to provide supporting experience from the conducted project.

3.5 Analysis and revisiting research questions

3.5.1 Assessing software development process understanding (RQ1)

With support from the EBR framework we decomposed students' arguments to a degree that allowed us to pinpoint the weaknesses of their development process model understanding. Looking at the frequencies in Table 3.4, we can observe that:

- For both years, a relatively large number of standalone argument components were identified (e.g. single premise, rule and data components in 2013 and evidence components in 2012). A standalone argument component indicates a lack of a coherent discussion, a concatenation of information pieces that does not create a valid argument for a specific claim. There are exceptions, e.g. PM#8

Table 3.4: Frequencies of identified argument components and argument content strength in project post-mortems for choosing a particular process model.

Year	Premise	Premise-Rule	Premise-Evidence	Premise-Rule-Evidence	Rule-Evidence	Rule	Evidence	Data	Evidence-Data	Premise-Data	Rule-Data	Rule-Evidence-Data	Premise-Rule-Data	Premise-Evidence-Data	Premise-Rule-Evidence-Data	**Unsound Argument**	**Weak Argument**	**Strong Argument**
2012	5	2	3	4	14	11	16	4	8	4	2	3	0	1	2	17	14	7
2013	30	5	5	9	17	20	9	16	9	3	9	6	2	1	0	12	42	14

and PM#24 (see supplementary material), which is also expressed in a strong argument content rating.

- Looking at the argument component combinations that indicate a strong argument (i.e. that contain a premise, a rule, and evidence or data), we can observe that assignments containing weak arguments outnumber assignments with strong arguments in both years.

The detailed analysis of arguments with the EBR framework could also help to create formative feedback to students. For example, in PM#5, the student reported 7 data points on the use of SCRUM but failed to analyze this data, creating evidence (describing relationships of observations) and connecting them to rules. On the other hand, we identified several assignments with the premise that the requirements in the project are unknown or change constantly (using this premise to motivate the selection of an Agile process). However, none of the assignment reports data on change frequency or volatility of requirements, weakening therefore the argument for choosing an Agile process.

Given these detailed observations one can make by applying the EBR framework we think it is a valuable tool for both assessing arguments and to provide feedback to students. However, this power comes at a price. We recorded coding times between 10 and 30 minutes per question, excluding any feedback formulation that the student could use for improvement. Even if coding and feedback formulation efficiency could be increased by tools and routine, one has to consider the larger effort this type of assessment requires compared to other means, e.g. rubrics [1].

3.5.2 Impact of SPSM on students' understanding of software development processes (RQ2)

The only difference in how the course was conducted in 2013 compared to 2012 was the use of SimSE simulation based software process games. Besides the limitation of this study (as discussed in Section 3.3.4) improvements in the students' understanding can be seen as indications of usefulness of SimSE based games for software process education.

In order to evaluate students' understanding, we measured the quality of argumentation for the choice of the software process. Concretely, we evaluated the content of the student reports by using the strength (classified as strong, weak and unsound) of an argument as an indicator. To test the hypotheses stated in Section 3.3.2, we used the chi-square test of independence [23] and rejected H_0 at a confidence level of $\alpha < 0.05$.

For the project management plan (Table 3.3), which was a group assignment and was delivered at the beginning of the course, the observed frequency of strong, weak

and unsound arguments did not differ significantly between 2012 and 2013. Hence we cannot reject H_0 for the project management plan.

For the project post-mortem (Table 3.4), which was an individual assignment and was delivered at the end of the course, the observed frequency of strong, weak and unsound arguments did differ significantly between 2012 and 2013 ($chi-squared = 8.608$, $df = 2$, $p-value = 0.009$). Hence we can reject H_0 for the project post-mortem and accept that there is a difference in process model understanding of students in course instances 2012 and 2013. However, since this difference only materialized at the end of the course, after the project has been conducted, the improved understanding cannot be attributed to the software process simulation game alone as discussed in the limitations of this study (Section 3.3.4).

Another indirect observation that shows a better understanding of the process model is reflected in the choice of the process model in the context of the course project (with small collocated teams, short fixed duration for project etc.). Compared to 2012 where most of the groups took plan driven, document intensive process models (two groups chose an incremental development model, one group chose Waterfall and only one chose Scrum), in the year 2013 all groups chose a light-weight, people centric process model that is more pertinent to the given context.

3.6 Conclusion

The EBR framework enabled decomposition of arguments into distinct parts, which ensured an objective evaluation of the strength of the arguments in student reports. This assessment allowed us to gauge students' understanding of software development processes.

The indications reported in this study (from use of software process simulation in an active course) adds to the confidence in evidence reported in earlier empirical studies in controlled settings. Given the potential gains as seen in this study, and relative maturity, user interface and decent documentation of SimSE, the minor additional cost of including it in a course to reinforce concepts already learned was well justified.

As future work, we intend to do a longitudinal study where more data is collected over the next few instances of the course.

References

[1] S. Barney, M. Khurum, K. Petersen, M. Unterkalmsteiner, and R. Jabangwe, "Improving students with rubric-based self-assessment and oral feedback," *IEEE Transactions on Education*, vol. 55, no. 3, pp. 319 –325, Aug. 2012.

[2] K. Beck, "Embracing change with extreme programming," *Computer*, vol. 32, no. 10, pp. 70–77, Oct 1999.

[3] J. Biggs and C. Tang, *Teaching for Quality Learning at University: What the Student does*, 3rd ed. Mcgraw-Hill Publ.Comp., Nov. 2007.

[4] J. B. Biggs and K. F. Collis, *Evaluating the quality of learning: the SOLO taxonomy (structure of the observed learning outcome)*. Academic Press, 1982.

[5] N. J. S. Brown, E. M. Furtak, M. Timms, S. O. Nagashima, and M. Wilson, "The evidence-based reasoning framework: Assessing scientific reasoning," *Educational Assessment*, vol. 15, no. 3-4, pp. 123–141, 2010.

[6] J. P. Campbell, V. A. Maxey, and W. A. Watson, "Hawthorne effect: Implications for prehospital research," *Annals of Emergency Medicine*, vol. 26, no. 5, pp. 590–594, Nov. 1995.

[7] A. Drappa and J. Ludewig, "Simulation in software engineering training," *Proceedings of the 22nd International Conference on Software Engineering - ICSE '00*, pp. 199–208, 2000.

[8] R. A. Duschl, "Assessment of inquiry," *Everyday assessment in the science classroom*, pp. 41–59, 2003.

[9] S. Erduran, S. Simon, and J. Osborne, "TAPping into argumentation: Developments in the application of toulmin's argument pattern for studying science discourse," *Science Education*, vol. 88, no. 6, pp. 915–933, 2004.

[10] A. Fuggetta, "Software process: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*. ACM, 2000, pp. 25–34.

[11] P. Hersey, K. H. Blanchard, and D. E. Johnson, *Management of organizational behavior: leading human resources*. Upper Saddle River, N.J.: Prentice Hall, 2001.

[12] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling : Why ? What ? How ?" *Journal of Systems and Software*, vol. 46, no. 2-3, pp. 91–105, 1999.

[13] G. J. Kelly and A. Takao, "Epistemic levels in argument: An analysis of university oceanography students' use of evidence in writing," *Science Education*, vol. 86, no. 3, pp. 314–342, 2002.

[14] A. Lawson, "The nature and development of hypotheticopredictive argumentation with implications for science teaching," *International Journal of Science Education*, vol. 25, no. 11, pp. 1387–1408, 2003.

[15] T. Lethbridge, J. Diaz-Herrera, J. LeBlanc, R.J., and J. Thompson, "Improving software practice through education: Challenges and future trends," in *Proceedings of the Future of Software Engineering, FOSE '07*, 2007, pp. 12–28.

[16] ——, "Comprehensive evaluation of an educational software engineering simulation environment," in *Proceedings of the 20th Conference on Software Engineering Education Training, CSEET '07*, 2007, pp. 195–202.

[17] D. Pfahl, O. Laitenberger, J. Dorsch, and G. Ruhe, "An externally replicated experiment for evaluating the learning effectiveness of using simulations in software project management education," *Empirical Software Engineering*, vol. 8, no. 4, pp. 367–395, 2003.

[18] A. Reznitskaya, L.-j. Kuo, M. Glina, and R. C. Anderson, "Measuring argumentative reasoning: What's behind the numbers?" *Learning and Individual Differences*, vol. 19, no. 2, pp. 219–224, Jun. 2009.

[19] D. Rodriguez, "e-Learning in project management using simulation models: A case study based on the replication of an experiment," *IEEE Transactions on Education*, vol. 49, no. 4, pp. 451–463, 2006.

[20] V. Sampson and D. B. Clark, "Assessment of the ways students generate arguments in science education: Current perspectives and recommendations for future directions," *Science Education*, vol. 92, no. 3, pp. 447–472, 2008.

[21] W. A. Sandoval, "Conceptual and epistemic aspects of students' scientific explanations," *Journal of the Learning Sciences*, vol. 12, no. 1, pp. 5–51, 2003.

[22] B. B. Schwarz, Y. Neuman, J. Gil, and M. Ilya, "Construction of collective and individual knowledge in argumentative activity," *Journal of the Learning Sciences*, vol. 12, no. 2, pp. 219–256, 2003.

[23] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures, Second Edition*, 2nd ed. Boca Raton: Chapman and Hall/CRC, Feb. 2000.

[24] S. E. Toulmin, *The uses of argument*. Cambridge, U.K.; New York: Cambridge University Press, 1958.

[25] C. von Wangenheim and F. Shull, "To game or not to game?" *IEEE Software*, pp. 92–94, 2009.

[26] A. Zapalska, D. Brozik, and D. Rudd, "Development of Active Learning with Simulations and Games." *US-China Education Review*, vol. 2, pp. 164–169, 2012.

[27] A. Zohar and F. Nemet, "Fostering students' knowledge and argumentation skills through dilemmas in human genetics," *Journal of Research in Science Teaching*, vol. 39, no. 1, pp. 35–62, 2002.

[28] *A Guide to the Project Management Body of Knowledge: (PMBOK® Guide)*, 3rd ed., ser. PMBOK Guides. Project Management Institute, 2004.

Chapter 4

Aggregating software process simulation guidelines: Literature review and industry experience

Abstract

Software process simulation is a complex task and in order to conduct a simulation based study practitioners require support through a process for software process simulation modeling (SPSM). Such a process should include what steps to take and what guidelines to follow in each step. This study provides a consolidated six-step process for SPSM where the steps and guidelines for each step are identified through a review of literature. The resulting process was used in an action research study at a Telecommunication vendor to conduct a simulation based study for the purpose of training developers. The experience of using the consolidated process is also reported for each step. We found the proposed guidelines to be sufficient for conducting a study in the case company for the given purpose.

4.1 Introduction

Software processes are complex, dynamic and non-deterministic in nature and it is therefore appropriate to study these using simulation. SPSM attempts to imitate the real-world software processes and enables investigations using the model for various purposes such as training, education, planning and strategic management [18] [14]. The potential of SPSM is well established and it is time to make an engineering discipline from this art form [34]. To achieve this various systematic processes to develop simulation models have been proposed in literature.

Today, when a practitioner takes on the task of SPSM the immediate challenge is the choice of a process for conducting an SPSM study because of the number of available process prescriptions in literature that claim successful results. These processes target certain simulation approaches e.g. Rus et al. [34] for discrete event simulation (DES) and Pfahl and Lebsanft [27] for System Dynamics (SD). Yet other differences in the proposed processes are the level of formalism in the process description [34], use of terminology or targeted experience level of modeler [3] and size of the organization undertaking SPSM [36].

Some of these proposals suggest that their process is extensible to simulation approaches other than the one they were originally proposed for, such as [34]. It is not obvious to an inexperienced modeler as to what part of the process will change if they use it for a different approach (e.g. using the process for SD to develop a DES model) or context (e.g. using the same process in a small enterprise). Similarly, the existence of these variations is confusing for a practitioner with no simulation background, because if the same process can be used why do we have a number of proposals? Therefore, it is important to analyze and use these processes to identify deficiencies and improvement opportunities. An aggregation of best practices and a consolidated simulation process will support wide spread adoption of software process simulation modeling in the software industry.

Ahmed et al. [1] found in a survey that software process simulation modelers consider the modeling process a major challenge in SPSM. They also present a possible explanation for the lack of an SPSM process discussion in literature, saying that simulation modelers do not report on the modeling process as either they already have a satisfactory process or are not interested in it. They [1] also found that most of the software process simulation modelers work alone in a simulation study. This may also suggest why the 'process' aspect is often overlooked as it will become more important in a collaborative study that requires communication.

Today, however, considering the amount of SPSM literature, the need is to combine the lessons learned and make use of the collective experience gained over more than three decades of SPSM research and practice.

The contribution of this study is three-fold:

1. First is the synthesis of a simulation modeling process from literature along with supplementary recommendations for each step of this consolidated process.
2. Secondly, this process, which utilizes experience of various simulation modelers (thus less influenced by individual preferences), is applied in an industrial simulation project.
3. Lastly, SPSM is a relatively new field, however, simulation for investigating various problems in different domains is an old and active research area. Thus, we also compared the results of this study to the research from venues outside the typical SPSM forums to see whether the process they use is really different from what we apply in Software Engineering. Such an attempt would identify the gap between SPSM and simulation research in general and indicates what can be learned and used between the disciplines.

The remainder of the chapter is outlined as follows. Section 4.2 summarizes the existing processes for SPSM and their industrial evaluation. Section 4.3 describes the research methods used in this study. Section 4.4 presents the aggregated guidelines and consolidated simulation process. Section 4.5 reports the experience of using the consolidated process in the case company. Section 4.6 reflects on the findings of the literature review and the action research. Section 4.7 concludes the chapter.

4.2 Related work

Pfahl and Günther [29] proposed a methodology, Integrated Measurement, modeling, and Simulation (IMMoS), which complements SD with existing modeling methods and quantitative measurement. One of the four components of this methodology is a process description for SPSM, which in turn has four phases and seventeen subactivities [27] [28]. Angkasaputra and Pfahl [5], analyzed this process against agile practices and made improvements to make it more agile. Zhai et al. [42] however claim that it is difficult to combine information learned from empirical studies into simulation results generated by IMMoS. Furthermore, they claim that it is difficult to build a SD model in practice using this approach, however, the paper does not state a rationale for that conclusion.

Rus et al. [34] create two clusters of the simulation process activities: engineering and management activities. They presented a process for DES, but also claimed that it can be used for other types of simulations. The process indicates its documentation-focus with emphasis on specification and approval of changes to the simulation model.

Müller and Pfahl [17] summarize five major steps for SPSM studies and refer to [29] and [34] for detailed guidelines for SD and DES respectively.

Ahmed et al. [3] proposed a process based on interviews of simulation modelers. This process was aimed at novice modelers and it was evaluated in a controlled experiment with students as subjects.

Silva Filho et al. [36] present a simulation approach that is targeted for small and medium sized enterprises. The process consists of these steps: The user selects the variables related to the subject to be addressed; Data collection; Preprocessing data; Causal model building; and Simulation model assembly and evaluation.

Kellner et al. [14] did not present a process for SPSM but identified major steps in a simulation study by answering the questions: Why simulate? (identifying the purpose and uses of SPSM); What to simulate? (the parameters and scope of simulation); and How to simulate? (techniques and approaches for SPSM).

Madachy [18] proposes the use of the Win Win Spiral model for model development and maps five major phases of a simulation study to this process. The book also provides comprehensive guidance especially focusing on use of SD in SPSM.

Park et al. [23] propose a three-step approach to develop a process simulation model based on Software & Systems Process Engineering Metamodel (SPEM). The three steps in their process are: Scoping (defining the simulation boundary in terms of portion of the lifecycle); modeling (defining the simulation model structure using UML, quantitative and qualitative models); and Transforming (algorithmic transformation of the models from previous step into DEVS-Hybrid simulation model). This proposal limits the undertaking of simulation to specific tools like the use of UML for modeling and DEVS-Hybrid for simulation of the model.

There is numerous industrial SPSM based studies however to the best of our knowledge, among the explicitly documented simulation processes only [29], [34] and [23] have been used in industry (all other industrial studies do not explicitly report the process they followed). In all three cases the original authors of the process description used it. The reason to mention this is that when one describes the process one followed a lot of the information may be implied for the authors and thus unconsciously missed from the description. Similarly, one may have consciously left certain details or decisions involved out of the process description considering them too obvious.

The multitude of similarity in these guidelines suggests that these approaches may be combined to develop a common process. Such a process will facilitate understanding, provide guidance and promote a systematic manner of approaching SPSM. It will also open new avenues for reuse. As having a shared process with identified activities and artifacts for SPSM will be the basis to identify how simulation models can be developed for and with reuse [22] and will help reduce effort and cost of the SPSM.

4.3 Research methodology

The aim of this study is to identify process descriptions for SPSM and the guidelines and best practices for their operational use. For this purpose following research questions were asked:
RQ1: *Which process descriptions and guidelines are reported for SPSM in industry?* The rationale for this question is to gather good practices in order to provide practitioners with a process that is based on accumulated knowledge of software process simulation.
RQ2: *Is the consolidated process (developed by synthesis of modeling and simulation processes from literature sources) useful in an industrial study?* The usefulness was evaluated from a practitioner's perspective that whether this process facilitates conducting a simulation study and enables development of a simulation model for the given purpose. Furthermore, given the limited experience reported from applications of the processes for conducting SPSM studies in industry, we complemented RQ1 by putting the process to use in the case company.

4.3.1 Literature review

The existing systematic literature review by Zhang et al. [44] [43] has covered the typical venues for SPSM. Therefore, we did not repeat a search for the time period already covered by this review. Also given the empirical evidence of effectiveness of snowballing versus database search [13] [12] we decided to supplement the search results of Zhang et al. [44] [43] studies by forward and backward snowballing. We used the Webster and Watson [39] three-step process in this study:

Step 1: Zhang et al. [44] [43] had reported a two phase systematic literature review scoping the research on the SPSM. They had reported [44] manually reviewing the key venues in the SPSM research, like the Conference on Process Simulation ProSim, International Conference on Software Process, special issues of Journal of Systems and Software and Software Process: Improvement and Practice [1] in the first phase. In the second phase [43] included more venues in manual search and complemented it with an automated search in primary research databases.

Therefore, in the first step we started with the primary studies from Zhang et al. [44] [43] review and identified articles relevant to the aims of our study.

[1] Since 2010 this journal has been incorporated in Journal of Software Maintenance and Evolution: Research and Practice.

Step 2: We reviewed the citations in the articles (backward snowballing) identified in Step 1 to find any prior articles that may be of relevance for the current review.

Step 3: We reviewed all the citations to the articles (forward snowballing) identified in Step 1 and 2 above. This was deemed necessary to identify any newer work on the topic.

Unfortunately, the classification of articles relevant to the methodology was not available for the second phase of Zhang et al. [43] study. Therefore, in Step 1 like in Step 3 we resorted to reading the titles and abstracts to select the potential primary studies. For Step 2 the discussion of a citation in the primary studies and its title were considered enough to make a decision about inclusion of a study as a potential primary study.

4.3.2 Data analysis

Full-text of the articles identified above was read and process steps were extracted. The statements representing these steps were copied verbatim in a spreadsheet and analyzed using the following three steps:
Step-1: For the first statement in the spreadsheet create and log a name for the process step (step-name).
Step-2: For each subsequent statement identify if a relevant step-name already exists. If it does log the statement with the same step-name, otherwise create a new step-name.
Step-3: Repeat Step-2 until the last statement has been cataloged.
A short description was given to each of the resulting groups with the same step-name. The first author did all the data analysis but by maintaining traceability between the resulting groups and individual steps extracted from the articles, the second author was able to review the correctness of resulting process steps.

Using this bottom up approach the steps in the consolidated process evolved based on the grouping. This reduced our influence on the formulation of the overall consolidated process.

4.3.3 Action research

We followed the action research methodology that is based on McKay and Marshall's work [20] depicted as six phases marked as A, B, C, D, E and F in Figure 4.1. In the remainder of the section we provide details of each of the phases of the action research in this study.

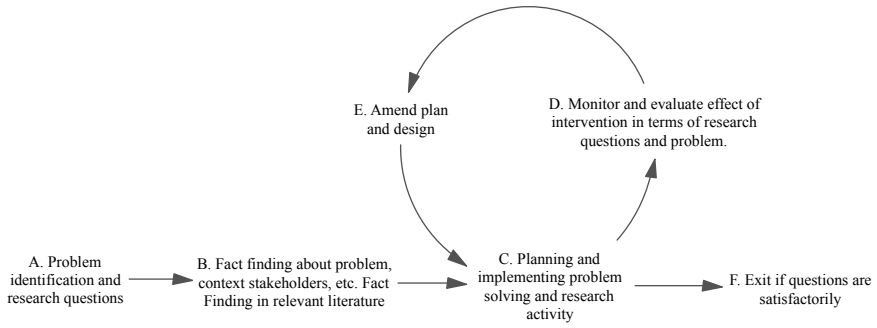


Figure 4.1: Overview of action research method (based on [20]).

A. Problem identification and research aims: An action researcher is interested in investigating a real world problem and generate new insights and knowledge during the process [20]. Therefore, we started with an introductory meeting and a follow up presentation to explain the potential uses of simulation, general steps in a simulation based study and tentative level of support and access to information required for the study. With this information the managers from the company identified a real-problem in development, "need for a better mechanism to illustrate and educate developers to realize the importance of early integration". We will explain the problem in detail in Section 4.4.1.

B. Understanding problem context and finding relevant literature: In follow-up meetings with four managers (one each from development and testing and two persons responsible for driving process improvement in the company) we characterized the problem identified in the previous step. We started to understand the industrial context by studying the testing process, identifying stakeholders and interviewing them for capturing different perspectives and triangulating our understanding of the process.

The goal of the simulation study was to assist product management develop a better case for early integration in testing of large systems. The testing process was simulated with a particular focus on the system in which all the component systems are integrated. The choice of this system was motivated by the complex interactions and dependencies involved in its testing process making it appropriate for simulation.

As directed by the action research methodology we attempted to understand the context of the problem at this stage. We used interviews, process documentation, and defect databases to understand the testing process at the company.

Process documentation of the test process is obtained in order to capture the specified test strategies and test levels conducted at the company. This information was used for triangulation purposes as an additional source to the interviews. Furthermore, the documentation aided the researchers to gain better domain knowledge and familiarity with the test specific terminology used at the company. In addition, we consulted official templates for defect reporting (e.g. forms for documenting defect data).

Interviews Using semi-structured interviews we focused on how testing is done, including characterization of test objects at different levels, input used to derive test cases and tool support. Interviewees were not selected to get a representative sample of a population, but rather to have a diversity of perspectives. Test Managers who have the knowledge about their team members supported us in interviewee selection ensuring diversity and coverage of various test levels in the company.

Quantitative data: With respect to defects we looked at types and distribution of defect. The data is available through a company internal and proprietary defect reporting system.

The resulting understanding of the testing process and development context at the company are briefly discussed in Section 4.5.

Similarly, faced with the task to develop a simulation model, we turned to literature for guidance on how to conduct a simulation based study. This lead us to pose research question RQ1. The analysis of findings, a number of proposed processes, from the literature review resulted in a consolidated process for conducting a simulation study (see Section 4.4).

C. Planning and implementing the problem solution and conducting the research: We applied the consolidated process while taking on the role of analysts doing process simulation in the company. We were working in the case company and had frequent informal discussions. However, the formal communication in this phase consisted of the following (where each meeting and interview lasted between 45-60 minutes):

- A follow-up meeting for data source elicitation with two managers (one from testing and another from product excellence).
- For variables where objective data was not available we got triangular estimates for such parameters (3 experts from testing teams each responsible for one or more test levels in the case company was involved in this meeting).

D. Monitor and evaluate the intervention from research and problem perspective: We documented our progress and reflections throughout the study in applying the consolidated process. In terms of solution to the problem we had two separate meetings for model demonstration. At this occasion we used a semi-structured interview [32] for model validation and initial feedback on the model's ability to achieve the training

goals. This allowed to have a planned interview guide, but also follow up on interesting discussion points.

In this study, we did only one iteration of the action research method. This iteration was used to reflect on the experience of using the consolidated process for creating the simulation model, this is the prerequisite to be able to build and engineer models that could actually later be used for the given purpose. The evaluation results from this iteration are used in a follow-up study where we want to evaluate the simulation model for its intended purpose in use (dynamic validation [11]).

4.3.4 Validity threats

In this section, we discuss the threats to the credibility of the findings of this study and what measures were taken to alleviate them. *Reliability:* The validity threats to the reliability of a study are related to the repeatability of a study i.e. how dependent are the research results on the researchers who conducted it [33].

In the literature review, data extraction from primary studies and analysis of the extracted guidelines were only done by the first author. This means that there is a risk of bias and correctness of results. However, to reduce this threat, the second author reviewed the extraction and analysis, and with that increased the reliability of the results.

Internal validity: The factors, which the researcher is unaware of or cannot control the extent of their effect, limit the internal validity of studies investigating a causal relation [33].

In the literature review, there is a risk of overlooking some relevant literature. We tried to minimize this risk by basing the study on a published systematic literature review. Using the guidelines [39] we started with an initial set of primary studies, we further supplemented the search by considering the citations in these studies (backward snowballing) and the citations to these studies (forward snowballing). Furthermore, snowballing has provided better results when compared to databases search in some studies [13] [12].

Inclusion and exclusion of articles was done by the first author alone therefore their is a threat of bias in selection. However, we tried to reduce the bias in selection by having explicit objective criteria. The systematic literature review [44] that was used as the input for our selection of software process simulation guidelines has the same limitation.

Both authors were involved in interviews, presentations and review meetings and compared notes and observations after each meeting. This reduced the threat of misinterpretation of feedback by the researchers.

In action research, instead of relying on a single perspective we involved multiple practitioners having different roles in the company to avoid any bias. Thus, through

triangulation using multiple practitioners from different teams we tried to avoid taking a biased perspective of the testing process.

Construct validity: The threats to construct validity reflect whether the measures used really represent the intended purpose of investigation [33].

Given that both authors had no previous experience of software process simulation modeling, we believe that we ideally reflected the situation of a practitioner who is faced with this task. This lack of prior knowledge reduced the likelihood that the successful outcome of this study was not an outcome of the consolidated process, but the expertise authors had in the area. However, there could be various confounding factors that we could not control in this study. There is a threat of maturation that we acquired more knowledge beyond what is expressed in these guidelines after reading considerable literature on the topic.

A majority of the process guidelines have been developed and used for SD based SPSM studies. There is a high likelihood that the consolidated process is biased in support for use of this approach. Furthermore, in this study because of various reasons (as discussed in Section 4.5) SD was the approach of choice. This means that the usefulness of the consolidated process needs to be evaluated for other simulation approaches. Having said that given the following reasons we consider it highly likely that the process will be useful for other simulation approaches:

- The similarity of the consolidated process and the process mainly used in quantitative discrete event simulation (presented in Section 4.6).
- All the contributing primary studies used to synthesize this consolidated process claim generalization to other simulation approaches.

External validity: The threats to external validity limit the generalization of the findings of the study outside the studied case [33].

The consolidated process has only been executed with a specific goal of training, using an SD approach for SPSM in the case company. This limits the generalization of conclusions beyond this specific case and simulation approach. As described in Section 4.6 the guidelines need to be adapted according to the purpose of the study. Therefore, this process needs to be evaluated for more purposes, using other approaches for SPSM in different contexts. Given this limitation of the study, using the description of company context and testing process (in this study) other companies in a similar context are likely to find the results transferable to their context [25].

Similarly, the current evaluation of the resulting model from use of this process can be only be considered as static validation [11]. The usefulness of the simulation model needs to be evaluated further for the ability of the model to achieve the intended learning objectives for developers. Indirectly, such an evaluation will provide important insights for improving the consolidated process that was used to conduct the study.

4.4 Consolidated process for SPSM based on literature sources

A common trait in all prescriptions of SPSM processes found in this study is the iterative and incremental nature of the process [3] [17] [18]. The consolidation of simulation processes in literature resulted in a process with six steps and it is presented in Table 4.1 and Figure 4.2. We present a brief description of each step, the sources which recommend them, the model heuristics [18] and guidelines applicable for each of the steps (in Sections 4.4.1 to 4.4.6). We also report our experience from using this process in an industrial study in Section 4.5.

Table 4.1: Mapping of the steps in the consolidated process to the contributing literature.

Process step	References
Problem definition	[14], [17], [34], [3], [18] and [28]
Model design	
Model Scoping	[14], [3], [23] and [18]
Identifying input parameters	[14] and [34].
Identifying result variables	[14], [34], [23] and [3]
Specifying reference behavior	[17] and [18]
Conceptual modeling	[14], [17], [34] and [23]
Choosing a simulation approach	[14]
Assessing technical feasibility	[28] and [3]
Implementation of an executable model	[17], [34], [23] and [3]
Verification and validation	[14], [34], [3], [17], [28] and [18]
Simulation based investigations	[17], [3] and [18]
Presentation and documentation	[3], [28] and [18]

The following are some guiding principles that are applicable in general to an SPSM study:

- Start small and later on add more content to the simulation model, look for analogies rather than starting the model building from scratch and work over an extended period of time instead of developing the model in one go [1] [4].
- Involve and maintain frequent contact with all stakeholders throughout the study [1] [4] [21].
- No model is perfect (i.e. it is an imitation and has a trade off between complexity and realism) [18].

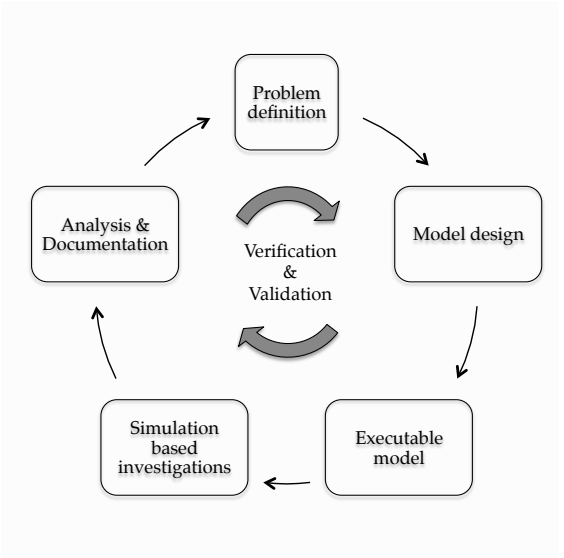


Figure 4.2: Consolidated process for conducting an SPSM study.

- All models are incomplete (as the goal is to model the behavior under study that is necessary to answer the questions of interest) [18].
- In industrial simulation studies it is important to deliver results and recommendations quickly as the modeled system and its environment are likely to change [4] [21].
- A model is not reality (therefore all results should be critically analyzed) [18].
- It is possible to build many different models of a single process (the perspective of stakeholders, level of detail and assumptions will make the model of the same process look very different) [18].
- Continually challenge the model (the credibility of the model can only be increased through further verification and validation V&V) [18] [4].
- The models are there to support managers in decision making, not to make the decisions for them [18].
- To facilitate effective communication use simple diagrams to communicate with others until they seek more detail (all stakeholders may not understand the equations and it may not be necessary to present those details) [18].

4.4.1 Problem definition

The starting point for SPSM is the identification of a problem to be investigated [14] [17] [34] [3] [18] [28]. The goals of SPSM influence important simulation decisions like the scope of simulation, level of detail, requirements on accuracy, which will in turn determine the required level of validation [30]. The common purposes for SPSM are [14]: Strategic management; Planning, control and operational management; Process improvement and technology adoption; Understanding; and Training and learning. The aim in this step is to identify the key questions that need to be addressed.

- Identify the users of the simulation model [28] [3].
- Define the usage scenarios (the use cases for the simulation model e.g. for the question: "How does missing the testing hand-off affect the cost and schedule of the release?", a corresponding scenario would be: "Keeping all input constant, miss the internal hand-off between testing phases for different types of requirements and observe the values for cost and duration for each of the cases.") of the simulation model [34].
- Develop test cases that clarify the model's purpose and will be later used for validation of the simulation model itself and its results [34].

Guidelines:

- Consider the audience, desired policy set (e.g. company rules such as having a certain level of quality before moving to the next development stage), and level of detail of the implementation when defining the model purpose [18].
- A model is created to answer specific questions [18], i.e. the SPSM undertaking should be goal-driven [27].
- Identify the users of the simulation model to allow their involvement in the simulation process. It will ensure that their needs are met and increase the chances of adoption of the simulation model in practice [27].
- A clear, well-specified purpose will guide the remaining process. It will also help avoid misuse of the model in the future [27] [18].

4.4.2 Model design

This step has six sub-steps where using the problem statement as a starting point we move closer to the implementation step. Decisions regarding the model scope, granularity, input and output are taken in this step. The major outcome of this step are a quantitative (mathematical formulation) and a qualitative (influence diagrams and causal diagrams) model of the system under study.

Model scoping

Depending on the aim of the study, the boundaries of the system being modeled are defined [14] [3] [18] [23]. The scope is defined in terms of the organizational breadth, product or project team, portion of lifecycle, single or concurrent projects and the time span that needs to be modeled. The system for one study may well be a subset of a larger system.
Guidelines:

- Independent of the scope of the study, to keep the customers interested, modeling should proceed iteratively ensuring the delivery of the first executable model as soon as possible [27] [21].
- The scope is largely dictated by the purpose hence the model purpose should be very focused to ensure that it does not become overly complex that in turn threatens its validity and credibility [27].
- Use GQM for defining goals, questions and the appropriate metrics for simulation [34] [18] [23].

Identifying input parameters

Identification of the input parameters for the simulation model is proposed in [14] [34]. Input parameters are the independent variables and are identified by assessing their affect on the system state an the choice of desired result variables (or study objectives).
Guidelines:

- Start designing the model early on, independently of the availability of data for calibrating the independent variables [18]. Also note that it is often not possible to measure all variables of relevance accurately.
- Select input probability distributions for input variables [7]. Law [16] provides a detailed discussion of statistical methods to support this decision.
- Ensure integrity of data by maintaining constant communication with stakeholders [21]

Identifying result variables

Identify the outputs required from the model [14] [34] [3] [23] to address the problem statement by answering the key questions identified earlier. Results variables can influence the process abstraction or level of detail because e.g. if instead of end-to-end process outcomes one is interested in intermediate values of a variable at various steps in the process then a certain granularity level of the model will be necessary.

Guidelines: Use of GQM to identify the output variables necessary to address the goals of the simulation [34] [18].

Specifying reference behavior

The specification of reference behavior is proposed in [17] [18]. Müller and Pfahl emphasize the importance of specifying observed problematic behavior or the desired behavior (called the reference behavior) [17]. It can also help to identify the model output parameters and can be used later as input for model validation.
Guidelines:

- The reference behavior should be defined (plotting behavioral change over time) [18] based on historical data [26].
- Consider behavioral patterns based on experience when actual data is not available in the company [18] [26].
- Instead of striving for absolute measures focus on relative measures (e.g. increase defect count in %) [18].

Conceptual modeling

Conceptual modeling is proposed in [14] [17] [34] [23]. In this step, process elements, information flows, decision rules and behavior that have an influence on the result variables and are relevant to the simulation goal are identified for modeling. This step involves making use of both the explicit and tacit knowledge about the software process. Some other activities in this step are as follows:

- Create static process models to understand the flow of information and transformation of artifacts in various activities [34].
- Create influence diagrams where (positive or negative) influence of various parameters on each other is depicted [34].
- Create causal loop diagrams (more pertinent for SD) [18] [28].
- Collect and analyze available empirical data for deriving the quantitative relations identified in the influence diagrams [34].
- Distinguish the parameter type as calibration, variable or project specific input parameter [34].

Guidelines:

- Consult domain experts as they have knowledge beyond the documentation and can judge the relevance of information to the problem under study [17].

- Interview domain experts during modeling to avoid missing important aspects of the process and reduce the threat of misunderstanding [27].
- Motivate the choice of cause-effect relations with data sources and literature [26].
- Do not try to model the "system", i.e. include only the entities and relations necessary to generate the behavior of interest. Using a top-down iterative approach, add details only when necessary [18]. Start small and later on add more content to the simulation model [1].
- Using the goal and the scope of model, aggregate and abstract to an appropriate degree hiding unnecessary details [18].
- Keep both the model and the communication with users simple [18]. Review each individual cause-effect relation in the causal diagram before the combinations [26].

Choosing a simulation approach

Choosing a simulation approach is proposed in [14]. As software processes have both discrete and continuous aspects, e.g. there are discrete quantities like lines of code, defects, number of requirements etc. and continuous aspects like staff experience, motivation level, commitment, cohesion in the teams etc. So different types of simulation approaches are applicable in SPSM [18]. The choice of an appropriate simulation approach will therefore depend on the aim of the study and the modeled system. A multitude of simulation approaches are available. Zhang et al. [44] found 10 approaches that have been used for SPSM. The most commonly used approaches are SD and DES. The choice is again influenced by the purpose, scope and result variables among others.
Guidelines: Continuous simulation is the approach of choice when the analysis does not require the low-level process details e.g. for strategic analyzes and long-term behavior. SD models have levels and flows of entities. These entities are not individually identified and are not traced through the process [14]. DES is the approach of choice for detailed process analysis, relatively short-term analysis. DES models contain distinct entities with attributes that move through the process [14].

Assessing technical feasibility

Establishing technical feasibility is proposed in [28] [3] i.e. prerequisites for simulation model development and usage like adequacy of problem definition, availability of data, and process maturity [26]. Before the simulation model development, technical feasibility should be checked [28] [3]. Ahmed et al. [1] found that most of the modelers assess feasibility as a first in the modeling process.

Guidelines: The feasibility aspects identified in [26] are covered by the following questions proposed by Balci [7]:

- Do the benefits of conducting a simulation based study justify the cost?
- Is it possible to use simulation for the goal of the study?
- Is it possible to complete the study in the given time and resource constraints?
- Is the necessary information available e.g. classified or not available?

4.4.3 Implementation of an executable model

Implementation of an executable model is the next step in the process as proposed in [17] [34] [23] [3]. The choice of simulation approach chosen for SPSM will determine the details of this step and exactly how the model will be implemented. Rus et al. [34] are the sole proposers of creation of "high level" and "detailed design" of the simulation model. None of the other process descriptions have suggested such a design activity. Ahmed et al. [4] in a survey also found that most simulation modelers do not produce a design prior to implementation of the simulation models.
Guidelines:

- Start with a small model. The entire simulation model should not be developed in one go rather create the model in an incremental manner [18] [1] [4].
- The use of relative measures and normalizing them helps in scaling the model. [18].
- Develop the simulation model with high modularity [1]. Separate data from model to support modification and experimentation [21].
- Always keep the model in a state that it could be simulated (or tested) [18].

Choosing simulation tools and techniques

Another important decisions is the choice of simulation tool as proposed in [14] [3]. A stochastic simulation requires multiple simulation runs to gain reliable results from statistical analysis of the result variables. A number of tools are available today to facilitate this analysis. Many come with graphical interface (facilitating walk-through), output visualization, support for interactive simulation, sensitivity analysis and connectivity with third party applications.
Guidelines: Madachy [18] and Ahmed et al. [4] list the price of the tool, ease of use of the tool, training opportunities, documentation and maintenance support, computer platform and user familiarity, and performance requirements of the simulation model as the criteria for choosing an appropriate simulation tool.

Model calibration

Model calibration using the actual data is proposed in [18] [14]. To facilitate data collection simulation model may be integrated with the domain systems [21]. However, a major issue than integration is the lack of data for model calibration and validation in real-world settings [14]. The desired data is often poorly defined, inaccurate or missing altogether.
Guidelines: To deal with a lack of project data Park et al. [23] recommend use of analytical models like COCOMO II [8].
Here are some suggestions to deal with the typical situations that modelers have to face:

- Consult domain experts to deduce the accuracy and relevance of data [21] [30].
- If the desired metric is not available but "a similar one is" then make calculated adjustment, make adjustments by consulting experts, use source documents if you need finer granularity instead of aggregates, adjust decision variables to capitalize on available data or adjust the model scope [30].
- If the desired metric is not available and there are no similar ones: reconstruct the metric using other data sources, estimate using expert opinion, look for data in literature or drop the variable altogether [30].

4.4.4 Verification and validation

Verification and validation of the simulation model had been proposed in [14] [34] [3] [17] [28] [18]. A simulation model should undergo validation to the extent possible [14] and V&V should go on throughout the simulation study [34] [4]. However, it is difficult to thoroughly validate a software process simulation model because the data required to do so is often not available and validation throughout the process is costly [30]. Before the full-scale development of the simulation model the following steps should be done:

- Validation of the SPSM requirements [34] identified problem against the formulated problem [7].
- Review of causal diagrams with domain experts [28].
- In the review meeting describe the problem that will be addressed with SPSM, delineated system boundary and what will be the expected output of the study [18]. Use presentations, reviews and meetings with the stakeholders to communicate any change in the model or data [21].
- Developing a prototype of the simulation model to show how a user of the model will manipulate the input and control variables in the simulation [18].

Guidelines: Seven V&V activities are presented in [30]. Ahmed et al. [2] propose a framework for evaluating software process simulation models, which may be used as a guide to decide appropriate means of V&V covering different quality aspects of a simulation model. For a good description of techniques and steps to develop a valid and credible simulation model see [16], [15] [6] [7]. Some guidelines from other researchers are:

- Validation should be ongoing as parts of model are developed [4].
- Assure face validity by using inspections and structured walk-through of the model [14] [34] [3].
- False expectations (e.g. perfect predictions on the first model run) should be avoided, rather patterns should be reviewed for qualitative similarity instead [18].
- Consider constraints from the real world when conducting sensitivity analysis for a policy or a combination thereof to make sure that the model reflects real world behavior [18].
- Comparing simulation output to system output (actual data) [3]. See [16] for statistical methods to perform this comparison.
- Model validity is a relative matter [18] i.e. depending on the purpose of the study.

Sensitivity analysis

Sensitivity Analysis has been suggested as a V&V method [14] [3] [18]. It is also used as a method to design scenarios to investigate the behavior of modeled system with extreme values. It is applicable to all types of simulation approaches [14]. It explores the effect of changing certain parameter values on result variables. It helps to identify how significant is the effect of a certain parameter on the results of the simulation. The extent of statistical analysis however can be adjusted according to the use of data and the model [30].
Guidelines:

- Madachy [18] discusses application of numerical, behavioral and policy sensitivity analysis on simulation models.
- Begin with changing values of one parameter at a time keeping others constant [18].
- Statistical analysis is facilitated by good interconnectivity (automated data transfer) between the simulation and statistics tool [38].

- For stochastic models use Monte-Carlo analysis [18].

4.4.5 Simulation based investigations

Simulation based experimentation is proposed in [17] [3] [18]. In this step, we design and conduct investigations to produce simulation results that will answer the key questions raised in the start of the simulation study. We purposefully replaced "experiments" with "investigations" as we agree with Wernick and Hall [40] that the use of the term "experiment" is misleading in the context of SPSM compared to its meaning in empirical research.

Guidelines: Raffo and Kellner [30] explored various analytical and statistical techniques to evaluate process alternatives using simulation results. They [30] also showed how techniques like Data Envelopment Analysis (DEA) [19] can be used to perform impact analysis of such decisions.

Law [16] provides a good discussion of statistical experiment design and confidence interval based analysis of alternatives.

4.4.6 Presentation and documentation

Presentation of the simulation results and documentation of the study and the model is proposed in [3] [28] [18]. To communicate the simulation results to the sponsors of the study, appropriate means of presentation need to be selected i.e. speaking the language they understand [7]. The examples of presentation in this case could be integration of simulation results with an existing system that the domain experts use or direct presentations using slides.

The documentation of the simulation study and the model will enable replication of the study and facilitate maintenance of the model. The simulation model, its goal and its usage is documented along with a discussion of simulation based investigations within the system's context. A proper documentation will ensure that the model is not misused in the future.

To summarize, the documentation should cover each of the steps proposed in the consolidated process in this study. For details of what is relevant and important to document in each step see the documentation template proposed by Madachy [18] and from scientific publishing perspective, please refer to França and Travassos reporting guidelines [9].

Guidelines: .

- Describing the goal and scope of the model [18] [9].
- Documenting causal relations and feedback loops [9]. Like any software, well-commented equations are easier to understand [18].
- Document assumptions, calibration values and rationales [9] [18].

- Verification and validation procedures and their outcome [9].
- Show output of multiple scenarios and discuss them [18]. The relevance and adequacy of these scenarios should also be motivated [9].
- Document any constraints and limitations in implementing a conceptual model in a simulation package [9].
- Like other empirical research, simulation studies should discuss the threats to study validity [9].

4.5 Experience of using the consolidated process

We conducted this study at a development site of a large Telecommunication vendor. The case and context are described to allow generalizing the results to a specific context. To develop an understanding of the testing process interviews with practitioners and process documentation were used. This facilitated in developing a simulation model that accurately captures the actual process. For readers, the description of the test process will help to understand the simulation model that was developed in this study (presented in Section 4.5.2).

The overall architecture of the product developed in the company consists of 12 systems, which are operationally independent and can also provide services independently of each other. The process used at the company is shown in Figure 4.3. In the first step the high-level requirements for the overall product are specified. Before the requirements are handed over to compound system development, a so-called "Go"-decision is taken, meaning that development resources are allocated to the high-level requirement. When the decision is positive, teams specify a detailed requirements specification, which then is handed over to the concerned system(s). The requirements are then implemented for a specific development system, and they are integrated (also called system level test).

The development is done in sprints run by agile development teams (AT Sprints in Figure 4.3). Each system can be integrated independently of another system, which provides them some degree of operational and managerial independence. However, the versions of two systems have to be compatible when they are integrated (Compound System Test). Each of the systems is highly complex, the largest system having more than 15 development teams. The size of the overall system of systems measured in lines of code (LOC) is 5,000,000 LOC.

Looking at other context elements [25] the following should be added as information:

- First versions of all systems are older than 5 years.

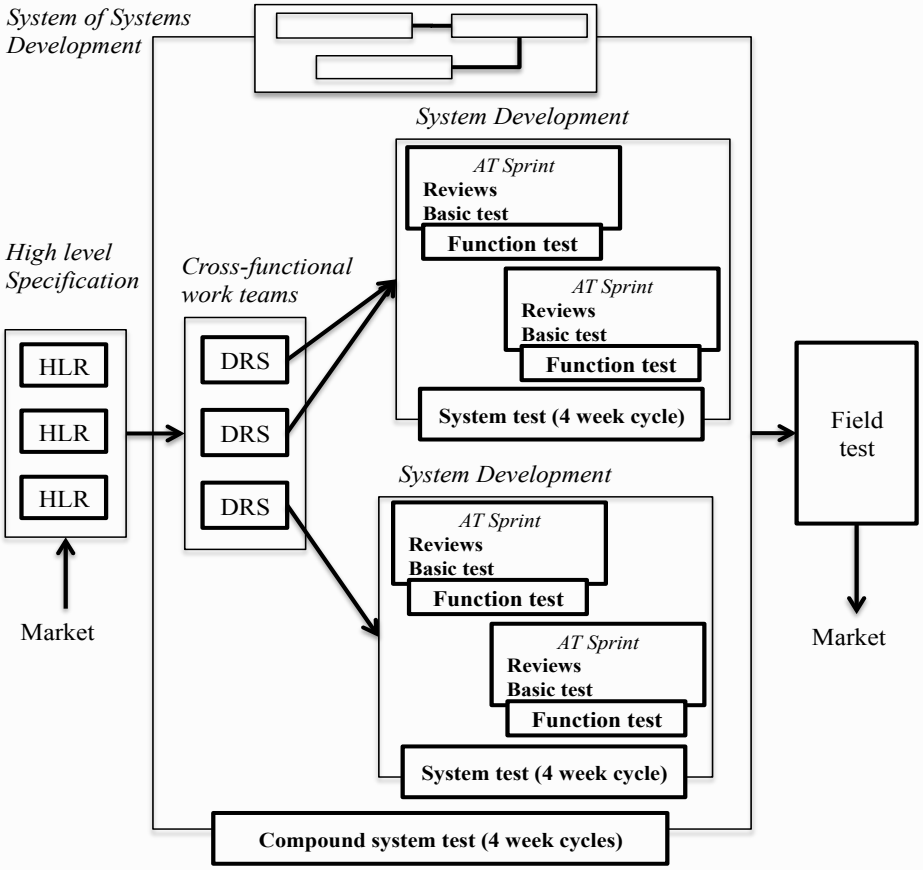


Figure 4.3: Development process with test levels at the case company.

- On principle level the development process is incremental with projects adding increments (e.g. new functionality) to the code base-line on system and compound system level.
- Within the teams and in the testing activities agile practices are used, such as: continuous integration, time-boxing with sprints, face-to-face interaction (stand-up meetings, co-located teams), requirements prioritization with product backlogs, re-factoring and system improvements

Overall there are six test levels (reviews, basic test, function test, system test, compound system test and field test) at the case company (as shown in Figure 4.3), however, for this simulation study we only focus on the "system test" and the "compound system test" levels. Reviews, basic and function test are abstracted in the development activity for this study and the field test is considered beyond the scope of this study as the product is considered ready for deployment and field test is just a release to limited customers.

4.5.1 Problem identification

We started the project with a presentation of the capabilities of process simulation and the main steps in developing a simulation model. We also emphasized the importance of having a focused and well specified goal for a simulation study. This helped convey the message upfront about the potential benefits of SPSM for the organization and the limitations.

The success of the initial presentation was also visible in the problem identified by the panel of managers from the company, as it was very focused and concentrated on limited phases of the overall development process. The goal and its sub-goals for the study are presented in Table 4.2.

In our case the initial set of users of the model was among the sponsors of this study. But after the initial prototype was presented, their confidence grew in simulation modeling and its potential. This gave us access to more end-users of the current simulation model (although with similar roles and responsibilities in the organization).

Table 4.2: Goal of the simulation study at the case company.

To develop a better understanding in developers of the impact of missing system-test iteration on the compound-test and the overall system.
- Demonstrate the impact of a requirement or a defect-report missing the system-test or compound-system test? - Demonstrate the impact of a requirement with dependencies missing the system-test or compound-system test? - Demonstrate the benefits of early integration?
The impact in above sub-goals was measured in terms of amount of rework, cycle time for tasks (requirements and defect reports), workload and productivity, and quality (number of failures).

As this SPSM study aimed at understanding and training of the managers, it was not reasonable to start a new top-down measurement program to support and achieve the goals of the study. We elicited a very focused goal and used Goal Question Metric (GQM) [37] only for documentation purposes. Instead of starting a new measurement program, we revised the goal and scope of the study to determine what questions can be answered given the metrics that are available, can be derived or accurately estimated.

We renegotiated the goal and scope of the study based on the priorities from the practitioners, existing metrics and trade-off on accuracy of results. Thus, the goal of evaluating the effect of the test strategy (which defects should be found at which test levels) was dropped due to lack of reliable fault slip through data.

Moreover, creating at least high-level usage scenarios helped to ensure that the modelers correctly interpreted the goals and questions that the model should be able to answer.

4.5.2 Model design

Model scoping

The problem statement identified earlier was discussed further in a subsequent meeting to elaborate and characterize the scope of the simulation model. It was decided to focus on the testing process at the company and understand the cost of delay when a requirement misses the system integration test cycle.

While deciding the scope of the system that is modeled it was important to see what level of dependence exists between the subsystem being modeled and the other systems that make up the compound system. In this case, it was the testing process that was the system of interest, but we could not capture the implications of decisions at this level without considering the overall development process. Therefore, we decided to have an abstract view of other processes and relatively more detailed view of the testing process.

The testing process was introduced in Section 4.5 and shown in Figure 4.3. Based on the purpose of the simulation study we decided to model the development activity at a higher granularity level where the development iteration included (implementation, basic testing and function testing). We were more interested in the test iterations of system level and compound system level testing. These decisions are visible in the conceptual model that is presented in Section 4.5.2.

Identifying input parameters

Even if an industrial practitioner takes on the task of SPSM, no one stakeholder has access to all the relevant information required. Also the accessibility of different stakeholders is another constraint in industrial settings. Therefore, we devised a template with these two considerations in mind. The template shown in Table 4.3, assisted us to effectively acquire the data required for calibration of the simulation model. While filling it out we noticed that certain metrics were not available, or that we have to consult other departments.

In cases where the information was not available this initiated a discussion with domain experts about the appropriate level of detail for simulation or acquiring estimates for the values. Doing the data source elicitation early on during the study helped to revise the goals of the simulation as well as the appropriate level of detail. For example, in our case the company was interested in seeing the effect of putting in more resources at a certain test level and how that reduces the required effort at other levels. However, the data to do this analysis was unreliable and abandoned. Since, this was

Table 4.3: Data collection.

Item	Description
Variable name	Name the variable e.g. Size
Description	A short description of the variable.
Variable type	Depending on how we want to analyze this variable as independent, dependent or control variable.
Stakeholders (roles)	The stakeholders responsible for this variable or who could give more information about this variable.
Activities	The activities in the software development life-cycle that will influence, produce or use this variable's values.
Software artifacts	Software artifacts where this variable can be computed from or is stored in.
Unit of measurement	The unit of measurement.
Simulation values	This may be different from the value above e.g. in the absence of exact numerical values we may have small, medium, large or intervals to estimate size for simulation.
Typical and extreme values	The typical, minimum and maximum value of this variable.
Distribution	The distribution of this variable.
Tolerance	The acceptable tolerance in the estimation, in case of a dependent variable.

not the highest priority issue for the stakeholders they decided to skip this as a goal for this simulation study.

Identifying result variables

Creating usage scenarios in "Problem identification" step (see Section 4.4.1) helped in identifying variables of interest. It also helped in categorizing them as input, output and control variables for the simulation model. For example, when we specified a usage scenario: What is the state of workload if the workforce allocation is kept constant and failure rate increases by 10 percent?

Specifying reference behavior

In the case company there was already a measurement based information visualization tool that can graphically represent the software process output. This was especially useful for us to document the existing behavior of the testing process. Apart from the benefits of reference behavior for validation and output parameter identification, we found that being able to replicate the reference behavior had a very positive impact on the interest levels of simulation-model users. As in our case the users were excited to see that the system behavior was replicated and gave credibility to the model. This increased interest was also confirmed by the fact that they enthusiastically took it up in the project management organization.

Conceptual modeling

What cause/effect relations should be modeled? This is a difficult choice as well and again one of the most influencing factors is the aim of the simulation. Other than the obvious "factors" e.g. rework positively reinforces workload, it is rather difficult for a beginner to anticipate more complex feedback loops. Looking at the existing simulation models was helpful to evaluate the relevance of various influencing factors and how others have mathematically modeled them. Had there been an aggregated collection we could easily choose which of these relations are important to map for the simulation goal at hand. The next difficulty in simulation is the level of abstraction at which to model the process so that there is a balance between the effort expended and realism depicted in the model. For a beginner it is not an easy decision. It was very helpful to use a top down approach, i.e. to start with an abstract representation and then add detail later. We did multiple iterations and if the customer thought that a certain aspect of the development process was over-simplified then we looked at how adding a

certain detail will effect the goal of the simulation. If we found that adding this detail is necessary to achieve the aims of the simulation, we incorporated it into the model.

We analyzed the empirical data early on in the study rather than waiting till the process modeling was done [34]. It helped to face the data availability issue early on and we revised the scope of the model through discussions with the stakeholders without spending any unnecessary effort. The conceptual model of the testing process in the case study is presented in Figure 4.4.

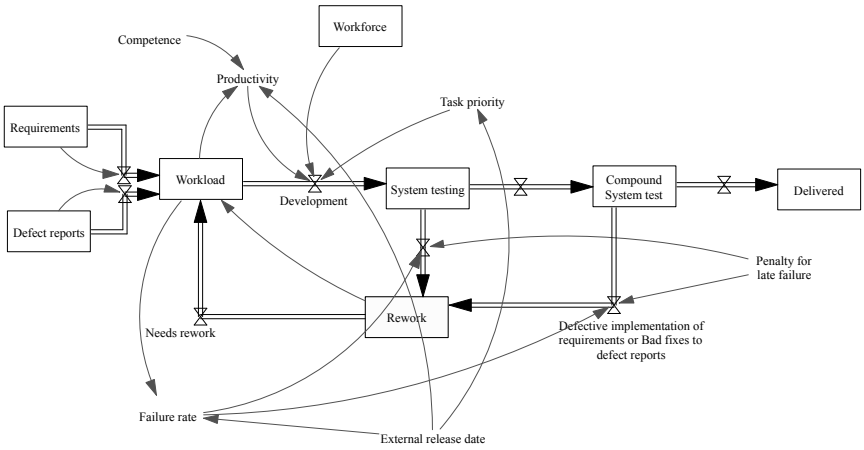


Figure 4.4: Conceptual view of the testing process at a high level.

Choosing a simulation approach

The decision of the simulation modeling approach is difficult for a practitioner with no simulation background. There is a long list of possible approaches that have been used for SPSM. We looked at literature and short listed SD, DES or Hybrid simulation with both continuous and discrete simulation because of the frequency of use and availability of tool support. By keeping the aims of the study in mind, we wanted to develop a better understanding of the process and use it as a training tool. We decided to use SD as it was sufficient to have a macro view of the process and model the overall flow of artifacts through the process. Furthermore, the general implication of "missing an iteration" (i.e. a requirement that has been implemented or a defect report that was

fixed has failed at a certain intermediate test level and has to go back to development and wait for the next test iteration) was more important for the users than what happens when a particular requirement "misses an iteration".

We felt that the case of doing hybrid simulation is very compelling as we could see the value of modeling both the discrete and continuous aspects of the process. In addition to the aforementioned reasons for starting with SD modeling, we are SPSM novices, unfamiliar with both discrete event and continuous simulation. However, we also decided to stay open to switch to a different choice later if we would be unable to achieve the goal with the current approach.

Assessing technical feasibility

As novice modelers we did not feel confident to comment on the "technical feasibility" before the initial model development as recommended by [28]. We tried to increase the chances of correctly assessing the technical feasibility by delaying the decision till initial prototyping of the model. The implementation is reported in Section 4.5.3 and we were able to develop the testing process simulation model.

4.5.3 Implementation of an executable model

We found follow-up meetings with the users very helpful where the implemented model was presented. The ability of the simulation tool to dynamically change values of parameters and show the effect on output variables was also very helpful for the users. It helped them to understand what was going on and reflect whether it made sense in their context.

Simulation tools and techniques

The choice of simulation tool, there is again a long list of simulation tools and the features they support. To the best of our knowledge there are no comparative studies where these tools may have been evaluated objectively for the learning curve, ease of use, effort required to model and interpret results. It will also be useful if we knew the bare minimum features that should be available and are useful for each simulation approach so that a practitioner may make an informed decision. We chose the Vensim tool because of the graphical interface, its ability to allow runtime changes to parameters, visual representation of result variables and state of the system. Some other tools also provided similar features, but as we chose SD and Vensim happens to be the tool of choice for this type of simulation [44]. This also motivated our choice.

Model calibration

The issues identified above and the mitigation techniques proved fairly comprehensive in our case. We faced most of the issues highlighted in Section 4.4.3 and chose relevant mitigation techniques. Some other challenges that we faced during calibration were:

- Always consult domain experts for the real meaning of the data fields even if they have a perfectly straightforward name. In the case organization, while eliciting the units of the variable "Size", we found that it was the estimated effort in person hours instead of a size metric. The template presented in Table 4.3 proved useful here to surface such misunderstanding before it could cause any major problems
- We found that there were certain "silos" of information e.g. the information about the requirements was in one database and defects reports in another and there is no explicit connection between them. For example, databases existed for both requirement and defect reports however the number of iterations that each requirement or defect report had to go through in testing before delivery was not available. So although both data points exist, we cannot use these as there is no traceability between them.
- During the discussion with domain experts we found that certain information in the databases was unreliable (although it had legitimate values). It was unreliable as it was difficult to compute in reality and was filled with typical guesses only because it is a mandatory field.
- We had to frequently find the right experts and consult them, which was fairly time consuming, since data required was distributed in various data sources across departments and it was difficult to find accurate descriptions for data-fields in the documentation.
- Another issue which was not as big as unreliable or missing data but still made the task of calibration difficult was the inability of various interfaces (that were custom built for a certain purpose) to export data to a file that may be consumed in other tools.
- The choice of the time span of historical data to use for the calibration of the model needs discussion with domain experts. As we have to use a time period long enough to model the true behavior rather than any temporal trend and yet capture the current reality. So the knowledge of underlying changes, e.g. if anything from the development process, programming language or development platform has changed then we need to be aware of that.

4.5.4 Verification and validation

We used a structured walk-through approach for face validity of the simulation model. We found that presenting more details in increments was helpful. Here is the order we followed in validation meetings:

- Repeat the goals of the simulation model.
- Discuss specific questions that will be answered by the simulation.
- Discuss usage scenarios [34] of the simulation model.
- Discuss the influence diagram.
- Discuss a high-level view of the model (hiding the implementation details) jump to the detailed version only if required.
- Discuss all assumptions and simplifications done in modeling the process and reasons for making them. The assumptions and simplification in this simulation model are summarised in Table 4.4.

We found developing concrete questions and usage scenarios was very helpful to validate that we (as modelers) properly understood the goal. A walk-through of influence diagrams and causal loops instigated a lot of discussion about which cause effects relations are visible in the organization's context. Stakeholders also reflected on how some cause-effect relations are strongly visible in their organization. It was interesting to see that contrary to our assumption that a higher schedule pressure would create a

Table 4.4: Major simplification in the simulation model at the case company.

Like any model this model has a lot of simplifications and abstractions, however we attempted to make a conscious decision about any simplification we do and document it. This will help later in interpretation of results as well.
Use of rates rather than individual attributes for simulation.
Beyond a certain level of workload increase all employees get demotivated and this results in loss of productivity (based on literature).
Only two groups: Experienced and inexperienced based on competence.
Competence of employees is depicted only with differing productivity rates (no impact on quality).
Employee attrition and learning is not modeled.
The prioritization from the management for development will be modeled by proportional allocation of resources to requirements and defect reports.
Requirements have been clustered in three groups, first requirements having short market window, second requirements with dependencies and large requirements and third small to medium requirements.

higher failure rate when the system is delivered to testing did not hold true in the organization. Understanding the reasons was beyond the scope of this study, but one reason may be the use of reviews, functional testing and automated test suites by development before the code is delivered for system and compound system integration testing.

We found that explicitly documenting and presenting the assumptions and simplifications was very useful. This helped get the stakeholders' perspective if these simplifications were unrealistic in their context.

We followed the presentation with a structured interview to assess the validity of the simulation model with practitioners having various roles in the organization and hence brought different perspectives of the software test process. The questions used in the interviews are listed in Table 4.5. We used face validity of graphical models, replication of trends in reference behavior, verification of model inputs and outputs (whether the calculations are correct) and qualitative assessment of reasonableness of model output. These were deemed sufficient, as the purpose of this model was understanding and training.

Since the purpose of the model is for demonstrating a particular phenomenon i.e. the impact of missing testing iterations, accuracy of results was not the highest priority. It was sufficient if the model manages to replicate the real life behavior and trends in output variables. Therefore, we developed a deterministic model of the process. For sensitivity analysis we only used three cases for each of the parameter values (minimum, typical and maximum). It was helpful to start with altering one variable at a time as it was convenient to isolate the effects of that and interpret whether it still made sense in all three cases or not.

Table 4.5: Questions used in the validation of the simulation model.

Some of the questions used in the simulation model validation meeting
Do you think the aim has been correctly interpreted?
Are the criteria (output and scenarios) to evaluate the aim of the study complete?
Does the model correctly represent the current testing process at the company e.g. no missing phase or inconsistent terminology?
Is the model missing a cause/effect relation or a feedback loop?
Are the simplifying assumptions reasonable and acceptable from their perspective?

4.5.5 Simulation based investigations

We have illustrated some of the different scenarios that can be demonstrated using the simulation model developed in this study. This provides a flavour of different educational aspects that can be highlighted using the simulation model. The results of four different simulation scenarios with hypothetical values are presented in Figure 4.5. The four example scenarios with different failure rates at each of the test levels are summarised below:

- Case 1: No failures at system test (ST) or compound system test (CST).
- Case 2: 10 % (DR & Req) fail in ST, 0% fail in CST.
- Case 3: 10 % (DR & Req) fail in ST and 20 % fail in CST.
- Case 4: 50 % (DR & Req) fail in ST and 60 % fail in CST.

Following values were kept constant:

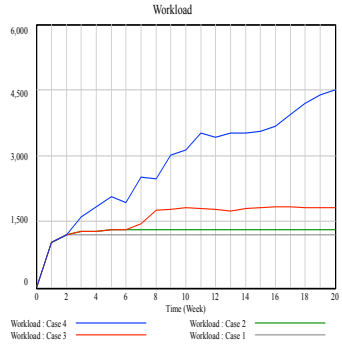
- Experienced workforce: 50 persons Productivity: 5-10 (Requirements (Reqs) or Defect Reports (DRs))/week
- Inexperienced workforce: 20 persons Productivity: 2.5-5 (Reqs or DRs)/week
- Workforce Allocation: 10 % for non-critical Reqs, 20 % for fixing DR and 30 % for requirements with dependencies and 40 % for Reqs with a short time to market.
- A constant influx of 1010 new Reqs & DRs

The effects of these failure rates are visualized on workload in Figure 4.5a, productivity in Figure 4.5b and delivery to customer in Figure 4.5c. The model was configured to lower productivity of experience staff when workload reaches a certain threshold. It can be seen that the productivity of experience developers drops in Case 4 (the blue line at week 5 and then again at week 7 in Figure 4.5b) and in this case the company does not even deliver to its full capacity (see the blue curve in Figure 4.5c).

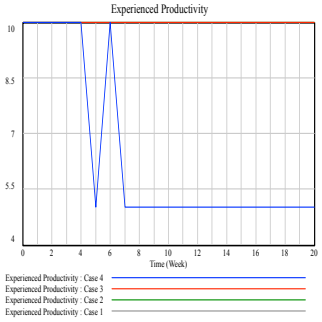
This model will be used to demonstrate that early integration results in improved flow of artifacts and consistent delivery to the customer.

4.5.6 Presentation and documentation

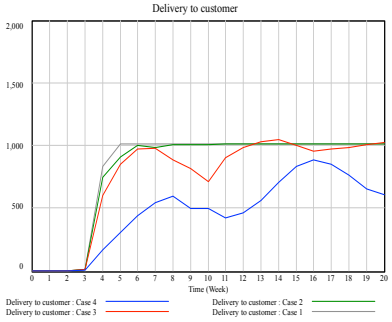
We used Madachy's [18] report template as a checklist when documenting the simulation model. We found that just presenting the values of result variables is not enough for the users. The results need to be put into context where the limitations of the results and their implications are interpreted and discussed. Since simulations are an abstraction of reality, the results were analyzed critically and other competing explanations for



(a) Workload in the simulated scenarios.



(b) Changes in productivity in the simulated scenarios.



(c) Overview of delivery to customer in the simulated scenarios.

Figure 4.5: Output of the simulated scenarios using different variables

the results were sought and presented to the users. This increased the credibility of the simulation results and initiated healthy discussions about the results of the simulation.

4.6 Discussion

We can see that the formalism that was earlier visible in the SE field in general also influenced the SPSM methodologies, where a lot of emphasis was on specifications, (see e.g. [34]). Most of the methodologies found in this study are incremental and iterative in nature, (see e.g. [3] [17] [18] [29]).

We have found that the overall process for SPSM is very similar and independent of the different simulation approaches, experience level of modelers, and the size of the organizations' where the study is conducted. By consolidating the individual process descriptions we have highlighted the important steps to systematically conduct an SPSM study. This process will increase the credibility of the resulting simulation model and the inferences drawn from its output. The only major differences occur in the implementation step, where the tactics and guidelines are particular to a certain simulation technique or tool e.g., [23] is useful for SPEM based simulation. The consolidated process in such cases contributes by highlighting what is still important e.g. V&V in case of [23] which may have been overlooked.

Owing to the general nature of the consolidated process we compared it with other fields which have an established use of simulation. The Winter Simulation Conference (WSC), since 1967, has become a premier forum for information on system simulation with a principal focus on discrete event, and combined discrete continuous simulation in all disciplines. We compare the simulation process presented by Shannon [35] in this forum (as shown in the second column of Table 4.6) to the SPSM-literature based consolidated process description (as shown in the first column of Table 4.6). We chose this as a comparison point as it is frequently cited (121 citations in Google Scholar) in diverse areas e.g. mining and healthcare. Compared to other engineering disciplines, absence of established physical laws raises unique challenges for model calibration and validation in SPSM. However, besides these differences, looking at the similarity between the two processes in Table 4.6) we may conclude that the overall simulation process is independent of the organizational context, simulation approach and experience of the modelers. The similarity between these two independently created processes also adds confidence to our literature based consolidated simulation process as it indicates the stability of the process.

Acknowledging the similarity of process and applicability of guidelines opens new possibilities to learn from other disciplines that have been using simulations far longer than relatively new SPSM discipline. For example:

Table 4.6: Major steps in a simulation study.

Steps in consolidated process for SPSM	Shannon's simulation process [35]
A. Problem definition B. Model design 1. Model scoping 2. Identifying the input parameters 3. Identifying the result variables 4. Specifying the reference behavior 5. Conceptual modeling 6. Choosing a simulation approach 7. Assessing technical feasibility C. Implementation of an executable model 1. Choosing the simulation tools and techniques 2. Model calibration D. Verification and validation E. Simulation based investigations F. Presentation and documentation	1. Problem definition 2. Project planning 3. System definition and determining the boundaries 4. Conceptual model formulation 5. Preliminary experimental design 6. Input data preparation 7. Model translation in a simulation language 8. Verification and Validation 9. Final experimental design 10. Experimentation 11. Analysis and interpretation 12. Implementation and documentation

- The work on V&V of simulation models by Balci [6] is useful. Similarly, Withers [41] has developed a mapping of best practices from software engineering to the practice of DES.
- The guidelines and tips given by Robinson [31] for the conceptual modeling activity, a vital part of a simulation study, are also valid and applicable for SPSM and have some overlap with Madachy's *"modeling heuristics"* [18]. Balci [7] presents a very similar lifecycle process for simulation studies (as the consolidated process presented here) and for each phase provides references to the literature outside the typical SPSM venues.
- Pawlikowski et al. [24] reviewed over 2200 articles on telecommunication networks and concluded that one of the two biggest issues leading to a lack of credibility in simulation based results is a lack of appropriate analysis of the simulation output. In a systematic review of simulation based studies in software engineering França and Travassos [10] found that only 2 out of the 108 articles

(reviewed in their study from an initial set of 946 articles) reported confidence intervals for simulation output analysis. Therefore, although the guidelines by Pawlikowski et al. [24] were based on the state of literature on telecommunication networks we think that their guidelines are equally beneficial for conducting and reporting the SPSM research.

Another indirect contribution of common generic process guidelines (such as the consolidated process presented in this chapter) will be to assist in identifying what is important from the documentation perspective and what needs to be documented to make these simulation based studies more credible.

4.7 Conclusion

This chapter presents a consolidated process for SPSM and presented the steps of the process and provided guidelines for each step. The process identified in literature is complemented by our own experience of using it in an industrial case. This led to the following answers for our research questions:

RQ1: *Which process descriptions and guidelines are reported for SPSM in industry?* We identified five different processes for SPSM. Although literature in SPSM differentiates between process descriptions for different simulation approaches but through analysis of these descriptions we found that the overall process of SPSM is independent of the simulation approach. The choice of a particular simulation approach will largely affect the specific operational tactics in the implementation phase of the SPSM life-cycle. Various authors have highlighted the lack of process descriptions for SPSM [34] [3] and then went on to propose simulation models to overcome this gap. Some differentiated the process models on the basis of formalism, level of experience of modelers and the size of the organizations undertaking an SPSM study. However, we found that there is a remarkable similarity between these proposals. It was also interesting to see that the combined process developed from consolidating these guidelines looks very similar to the simulation process ([35] and [7]) proposed in a venue that is not restricted to software domain.

RQ2: *Is the consolidated simulation process useful in an industrial study?* In our experience the consolidated process description is ample for overall design and execution of an SPSM study in industry. The consolidated process serves as a guide for systematically approaching SPSM in industry. However, the need is to supplement these guidelines with practical tactics relevant to the software domain, which may then be validated and improved with empirical evidence. We also found that the guidelines might vary with respect to the simulation goal. For example, our goal was training and

hence e.g. influenced the level of detail and how V&V was done, which would receive more focus when creating a model for project planning.

In future work the experiences obtained in this action research need to be extended by experiences in different model building contexts (e.g. models for prediction purposes).

References

[1] R. Ahmed, "Software process simulation modelling: A survey of practice," *Journal of Simulation*, vol. 2, no. 2, pp. 91 – 102, 2008.

[2] R. Ahmed, T. Hall, and P. Wernick, "A Proposed Framework for Evaluating Software Process Simulation Models," in *Proceedings of the 4th International Workshop on Software Process Simulation and Modeling (ProSim)*, 2003.

[3] R. Ahmed, T. Hall, P. Wernick, and S. Robinson, "Evaluating a rapid simulation modelling process (RSMP) through controlled experiments," in *Proceedings of the International Symposium on Empirical Software Engineering*, Piscataway, NJ, USA, 2005.

[4] R. Ahmed and S. Robinson, "Simulation in business and industry: how simulation context can affect simulation practice?" in *Proceedings of the 2007 Spring Simulation Multiconference*, ser. SpringSim '07. San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 152–159.

[5] N. Angkasaputra and D. Pfahl, "Towards an Agile Development Process of Software Process Simulation," in *Proceedings of the 6th International Workshop on Software Process Simulation and Modeling (ProSim)*, 2005.

[6] O. Balci, "Verification validation and accreditation of simulation models," in *Proceedings of the 29th Winter Simulation Conference*, ser. WSC '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 135–141.

[7] ——, "Guidelines for successful simluation studies," in *Proceedings of the 22nd Winter Simulation Conference*. IEEE Press, 1990, pp. 25–32.

[8] B. W. Boehm, *Software cost estimation with Cocomo II*. Prentice Hall, 2000, vol. 1.

[9] B. B. N. de França and G. Travassos, "Reporting guidelines for simulation-based studies in software engineering," in *Proceedings of the 16th International Conference on Evaluation Assessment in Software Engineering (EASE)*, 2012, pp. 156–160.

[10] B. B. N. de França and G. H. Travassos, "Are we prepared for simulation based studies in software engineering yet?" in *Proceedings of the IX Latin American Workshop on Experimental Software Engineering*, 2012.

[11] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice," *IEEE Software*, vol. 23, no. 6, pp. 88–95, 2006.

[12] T. Greenhalgh and R. Peacock, "Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources," *BMJ: British Medical Journal*, vol. 331, no. 7524, p. 1064, 2005.

[13] S. Jalali and C. Wohlin, "Systematic literature studies: Database searches vs. backward snowballing," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2012, pp. 29–38.

[14] M. I. Kellner, R. J. Madachy, and D. M. Ra, "Software process simulation modeling : Why ? What ? How ?" *Journal of Systems and Software*, vol. 46, 1999.

[15] A. Law, "How to build valid and credible simulation models," in *Proceedings of the Winter Simulation Conference*, vol. 37, no. 2. IEEE, 2009, pp. 24–33.

[16] A. M. Law, *Simulation modeling and analysis*. McGraw-Hill New York, 2007, vol. 4.

[17] M. Müller and D. Pfahl, *Simulation Methods*, 1st ed., F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer, 2008.

[18] R. J. Madachy, *Software Process Dynamics*. Wiley-IEEE Press, 2008.

[19] M. A. Mahmood, K. J. Pettingell, and A. I. Shaskevich, "Measuring productivity of software projects: A data envelopment analysis approach," *Decision Sciences*, vol. 27, no. 1, pp. 57–80, 1996.

[20] J. McKay and P. Marshall, "The dual imperatives of action research," *Information Technology & People*, vol. 14, no. 1, pp. 46–59, 2001.

[21] S. Murphy and T. Perera, "Key enablers in the development of simulation," in *Proceedings of the Winter Simulation Conference*, vol. 2, 2001, pp. 1429–1437.

[22] H. Neu and I. Rus, "Reuse in software process simulation modeling," in *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*. Citeseer, 2003, pp. 1–4.

[23] S. Park, H. Kim, D. Kang, and D.-H. Bae, "Developing a software process simulation model using spem and analytical models," *International Journal of Simulation and Process Modelling*, vol. 4, no. 3, pp. 223–236, 2008.

[24] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee, "On credibility of simulation studies of telecommunication networks," *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132–139, 2002.

[25] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *Proceedings of the Third International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, 2009, pp. 401–404.

[26] D. Pfahl, *An Integrated Approach to Simulation Based Learning in Support of Strategic and Project Management in Software Organisations*. Ph.D. dissertation, Fraunhofer-IRB-Verlag, 2001.

[27] D. Pfahl and K. Lebsanft, "Integration of system dynamics modelling with descriptive process modeling and goal-oriented measurement." *Journal of Systems and Software*, vol. 46, pp. 2–3, 1999.

[28] D. Pfahl and G. Ruhe, "IMMoS: a methodology for integrated measurement, modelling and simulation," *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*, 2003.

[29] ——, "IMMoS: a methodology for integrated measurement, modelling and simulation," *Software Process: Improvement and Practice*, vol. 7, no. 3-4, pp. 189–210, Sep. 2002.

[30] D. M. Raffo and M. I. Kellner, "Empirical analysis in software process simulation modeling," *Journal of Systems and Software*, vol. 53, no. 1, pp. 31–41, 2000.

[31] S. Robinson, "Conceptual modeling for simulation: issues and research requirements," in *Proceedings of the Winter Simulation Conference*, ser. WSC '06. Winter Simulation Conference, 2006, pp. 792–800.

[32] C. Robson, *Real world research: A resource for social scientists and practitioner-researchers*. Blackwell Oxford, 2002, vol. 2.

[33] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[34] I. Rus, H. Neu, and J. Münch, "A systematic methodology for developing discrete event simulation models of software development processes," in *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*, 2003.

[35] R. E. Shannon, “Introduction to the art and science of simulation,” *Proceedings of the Winter Simulation Conference*, pp. 7–14, 1998.

[36] C. Silva Filho, A. Cavalcanti da Rocha, and Others, “Towards an Approach to Support Software Process Simulation in Small and Medium Enterprises,” in *Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2010, pp. 297–305.

[37] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, “Goal question metric (GQM) approach,” *Encyclopedia of Software Engineering*, 2002.

[38] W. Wakeland, R. Martin, and D. Raffo, “Using design of experiments, sensitivity analysis, and hybrid simulation to evaluate changes to a software development process: a case study,” *Software Process: Improvement and Practice*, vol. 9, no. 2, pp. 107–119, 2004.

[39] J. Webster and R. T. Watson, “Analyzing the past to prepare for the future: Writing a literature review,” *MIS Quarterly*, vol. 26, no. 2, pp. xiii–xxiii, 2002.

[40] P. Wernick and T. Hall, “Getting the best out of software process simulation and empirical research in software engineering,” in *Proceedings of the Second International Workshop on Realising Evidence-Based Software Engineering*, ser. REBSE ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 3–.

[41] D. H. Withers, “Some fundamental issues in model building: software engineering best practices applied to the modeling process,” in *Proceedings of the 32nd Winter Simulation Conference*. San Diego, CA, USA: Society for Computer Simulation International, 2000, pp. 432–439.

[42] J. Zhai, Q. Yang, F. Su, J. Xiao, Q. Wang, and M. Li, “Stochastic process algebra based software process simulation modeling,” in *International Conference on Software Process*. Springer, 2009, pp. 136–147.

[43] H. Zhang, B. Kitchenham, and D. Pfahl, “Software process simulation modeling: an extended systematic review,” in *Proceedings of the International Conference on Software Process (ICSP)*. Springer, 2010, pp. 309–320.

[44] ——, “Software process simulation modeling: Facts, trends and directions,” in *Proceedings of the 15th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2008, pp. 59–66.

Chapter 5

Evaluation of simulation assisted value stream mapping: two industrial cases

Abstract

Value Stream Mapping (VSM) as a tool for lean development has led to significant improvements in different industries. In a few studies, it has been successfully applied in a software engineering context. However, some shortcomings have been observed in particular failing to capture the dynamic nature of the software process to evaluate improvements i.e. such improvements and target values are based on idealistic situations. The aim is to overcome these shortcomings of VSM by combining it with software process simulation modeling, and to provide reflections on the process of conducting VSM with simulation. Using case study research, VSM assisted by simulation was used for two products at Ericsson AB, Sweden. Ten workshops were conducted in this regard. Simulation in this study was used as a tool to support discussions instead of as a prediction tool. The results have been evaluated from the perspective of the participating practitioners, an external observer, and reflections of the researchers conducting the simulation that was elicited by the external observer. Significant constraints hindering the product development from reaching the stated improvement goals for shorter lead-time were identified. The use of simulation was particularly helpful in having more insightful discussions and to challenge assumptions about the likely impact of

improvements. However, simulation results alone were found insufficient to emphasize the importance of reducing waiting times and variations in the process. Due to the participatory approach followed in VSM, with involvement of various stakeholders, consensus building steps, emphasis on flow (through waiting time and variance analysis) and the use of simulation led to *realistic* improvements with a high *likelihood* of implementation.

5.1 Introduction

Enthralled by the success of lean development in manufacturing and product development [35] and the service industry [56], lean concepts have been adapted to the area of software development [46]. One of the practices in lean software development is value stream mapping (VSM) [46]. VSM is the practice of creating a value stream map that identifies the value added by each step in the software development process [58].

Starting with the *Current State Map* (CSM), this practice identifies *value-adding*, *required non-value adding*, and *non-value adding* activities in the current process. Making this distinction enables the improvement in the current process by the elimination of non-value adding activities [58] and creates a desired *Future State Map* (FSM) and an action plan for improvements.

The use of VSM has led to several tangible benefits e.g. McManus [34] reports studies where organizations were able to reduce the cycle time by 75%, reduced the standard deviation of the cycle times and significantly reduced the rework required to fix defects. It is also reported that typically up to 25% savings in engineering efforts are achieved with VSM based improvements [34].

Similar to manufacturing processes, in the development process for aerospace, information inventories had *"engineering work-packages"* inactive for up to 77% of the time [34]. The success of VSM outside manufacturing and production and in the design and development process of engineering makes it interesting to explore if it can be leveraged in software development.

Khurum et al. [29] adapted and applied VSM in the context of large-scale software intensive product development. They found the following two shortcomings based on practitioners feedback [29]: First, the notation used only provides a snapshot of the system and fails to capture the dynamic aspects of the underlying processes. This leads to a simplistic analysis to identify bottlenecks. Second, the improvement actions and the target value maps are based on idealistic situations. This also reduces the realism in the target value stream map (as it is based on ideal conditions that the identified problems are completely resolved).

These limitations reduce the confidence in the improvement actions identified in VSM, which implies that it is less likely that such improvement actions will be implemented. Given these limitations, as identified by Khurum et al. [29], and the potential benefits of software process simulation modeling (SPSM) [28], we argue that SPSM can be used to overcome these shortcomings. Other disciplines where VSM has been used have also seen a utility in combining VSM with simulation (cf. [31, 25, 5, 55, 8, 8, 53]). Some reasons for such combination in these disciplines are listed below:

- Simulation helps to reason about changing the process, and supports consensus building by visualizing dynamic views of the process [33, 1, 2, 19, 21, 25, 31, 47, 60, 62, 18]
- VSM alone is time consuming and simulation can assist to speed-up the analysis of the CSM, and the derivation and verification of a FSM [30, 49, 7, 18]. To verify the impact of changes and answering questions that cannot be answered by VSM [7, 5, 10, 11]
- VSM provides a snapshot, it is unable to detail dynamic behavior. Simulation can help predict the flow and levels and provide more accurate quantitative measures given its ability to handle both deterministic and stochastic inputs [33, 30, 53, 52]
- VSM alone cannot capture the complexity in terms of the iterations, overlaps, feedback, rework, uncertainty and stochasticity of the process [61, 63, 55, 32, 39, 30]

The contributions of this study can be briefly summarized as the following:

Contribution 1: Based on literature, this study proposes a framework to perform simulation assisted VSM.

Contribution 2: Evaluation of the use of simulation assisted VSM in industry.

To achieve the evaluation contribution, we conducted the study at a development site of Ericsson AB, Sweden. This entailed ten (three hours long) workshops for two products with five to six relevant practitioners in each workshop, and extensive quantitative and qualitative data gathering and analysis.

The evaluation of simulation assisted value stream mapping was done from three different perspectives, namely: *(a)* the practitioners participating in the activity, who are to benefit from and utilize the approach *(b)* the facilitators who conducted the simulation assisted VSM *(c)* an external observer, who is a simulation expert. The external observer assesses the process from an academic point of view. His expertise (having defined guidelines for simulation and prior hands-on simulation experience) allowed a critical observation of our use of simulation in a practical situation. He had a guideline focus compared to the practitioners, facilitators and our view as researchers doing

applied research using simulation in the context of VSM. Furthermore, we provide a detailed account of the findings to illustrate which wastes and improvements could be identified through this approach.

The remainder of the chapter is structured as the following: Section 5.2 presents related work. Section 5.3 maps the strategies for combining simulation with VSM and details the process for performing it. Section 5.4 describes the research method. Results are presented in Section 5.5. Section 5.6 discusses the results. Section 5.7 concludes the chapter.

5.2 Related work

Through an extensive search and by considering references that existed in secondary studies on lean in the software engineering (SE) context, we identified three primary studies reporting the use of VSM in software development [36, 27, 29]. While Yang et al. [61] have applied VSM to improve the lead-time of an IT company's R&D process for a hardware component. Mujtaba et al. [36] used VSM as a means to reduce the lead-time in the product customization process. Kasoju et al. [27] have used VSM to improve the testing process for a company in the automotive industry. Khurum et al. [29] have extended the definition of waste in the context of software intensive product development. While value should be considered fully from the perspective of the customer, they have added further dimensions potentially relevant for the customer that were traditionally not considered. However, these value considerations have to be evaluated on a case to case basis, based on what each organization conducting the VSM considers important from their customers' perspective. They also reported the process of conducting a VSM.

Khurum et al. [29], Kasoju et al. [27] and Mujtaba et al. [36] have used conventional static VSM notation. Yang et al. [61] have applied VSM in combination with simulation for a hardware product. To the best of our knowledge, this is the first study focusing on the investigation of simulation assisted VSM for software product development. However, from literature in manufacturing, production, service industry and product development five strategies were found to combine VSM with simulation, these are summarized in Table 5.1. In Section 5.3, we map these strategies on the process [29] for conducting VSM.

Table 5.1: Overview of research on the combination of VSM and simulation.

ID.	Strategy	Purpose	References
1	Developing simulation model for both the CSM and FSM of the system	Compare the CSM and FSM to demonstrate the benefits achieved by improvements.	[31, 25, 1, 19, 49, 60, 63, 32]
2	Develop simulation model for the FSM	To assess the effectiveness of the proposed improvements in terms of the intended benefits, and the likelihood of achieving those benefits.	[5, 10] [47, 11, 61, 39, 57] [62, 21, 26, 7, 18]
3	Develop a simulation model of the CSM	To identify or verify the bottlenecks in the CSM.	[2]
4	Develop simulation model to derive FSM	To evaluate several alternatives for FSM. To find optimum values for parameters like batch-sizes, work-in-progress, resource allocation and scheduling etc.	[2, 33, 38, 8, 9]
5	Use VSM based notation as a front-end for simulation model	To automatically generate simulation models from the notation used. This approach can achieve all the intended uses of simulation, in combination with VSM, as identified in the above strategies.	[52, 15, 55, 53, 30]

5.3 Framework for simulation assisted VSM (Contribution 1)

The challenges that motivated us to explore the use of simulation in conjunction with VSM are very similar to the motivations that researchers from other disciplines have had for this combination (see details in Section 5.1). This suggests that although we tend to consider the software development process more complex and dynamic [14], production processes have their share of these attributes as well.

Building on the research from other disciplines, we will map their strategies (see Table 5.1) to combine simulation with VSM to the VSM process as defined for software development [29]. Moreover, we provide useful references to SPSM research (from Chapter 2) that will guide the use of these strategies for conducting VSM with simulation for software product development. We also discuss the implications and likelihood of success of these strategies for VSM of software product development.

To successfully conduct a VSM in an organization the following three roles are required.

VSM Manager: a person who acts as the champion for the VSM activity [50] having authority to adapt the goal from upper management, get the commitment from participants for availability.

VSM Facilitators: one or more persons responsible for moderating the sessions, guiding discussions, taking notes, doing analysis of findings from the workshops and presenting the results. These people require a mix of knowledge and experience of lean and product development improvement expertise [34].

VSM Participants: stakeholders with knowledge and experience encompassing the end-to-end process having a variety of perspectives [34]

The VSM process has five distinct steps as shown in Figure 5.1. Five strategies (see Table 5.1) to assist VSM with simulation are placed appropriately in the VSM process (as shown in the Figure 5.1).

Except the first step that is handled in a meeting with a few key participants (as detailed in Section 5.3.1) all the remaining steps are done in a workshop setting with participants representing various perspectives from the current process. The workshop enables the participants to share the concerns and have informed discussions about challenges, underlying reasons, potential impact of change in one aspect on other phases of the process, in short this allows us to take a consensus driven approach to end-to-end process improvement [29]. For more details on each step and some useful templates for data collection please refer to Khurum et al. [29]. A summary of the activities, main purpose, roles involved, data sources used, and types of analysis done

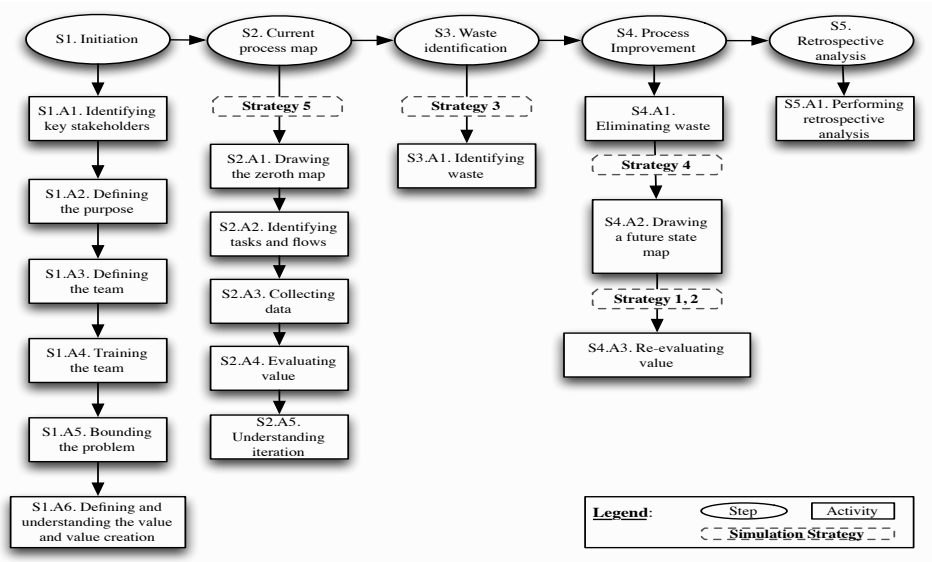


Figure 5.1: Overview of the VSM process and potential points for the use of simulation.

are presented in Sections 5.3.1 to 5.3.4 along with the additional information on how simulation can support each of these steps.

5.3.1 Initiation

The first step S1 (see Figure 5.1) constitutes the prerequisite activities that are required to support and successfully complete a VSM in an applied setting. The focus is on getting support from the relevant stakeholders and to engage the right persons in the activity. Furthermore, the scope of the process to be investigated (start- and end-points) is determined.

5.3.2 Current process map

In the next step S2 (see Figure 5.1), an attempt to understand the current value stream is made. This step aims to go beyond the documented process and really make explicit the current way of working. This step is essential to establish a current state map, as in practice often the processes are poorly documented, which may refer to obsolete practices, contain irrelevant details or the details that were once critical but have become irrelevant over time [34].

Although it is not mandatory, VSM typically uses a specialized notation to draw the current map [34, 50]. However, in previous experience [29], it was found that practitioners found it more intuitive to represent the process in the way they perceive and experience it. Use any existing process documentation to gain confidence in the correctness of the current process map. The point is not intended to show that the documented process is not followed, but to establish as good a baseline as possible of the current practice in the current process map. This is important as the current process map serves as the foundation for the rest of the VSM activities.

Simulation assistance: Literature on VSM outside SE, suggests the use of a notation that can be used to automatically generate simulation model of the current state map (see Strategy 5 in Table 5.1 and the related literature, such as [52, 15, 55, 53, 30]). Theoretically, this is the most appealing strategy as once this automatic translation is achieved, there is no additional cost of building a simulation model and one gets all the benefits of having a dynamic model. In SPSM, work by Hsueh et al. [24] is very pertinent. They suggest the use of a UML based notation to document the software development and then generate simulation models from it. Once a dynamic model of the process is available, it can be used for all the various purposes as listed in Table 5.1, e.g. to identify bottlenecks in the current state map. The incoming requests' pattern can be changed to simulate the increase in enhancement requests, mimicking the increased load on the process due to the product's launch in a new market.

However, not many companies may fulfill the prerequisites of having UML models of their processes as a baseline [45]. Thus, this very promising strategy to have process models in UML that can be simulated, may not be practical in a majority of the companies. For others interested to explore this strategy of combining simulation, Hsueh et al.'s work from SPSM [24] should provide a starting point.

Another limitation of the proposals of using UML to capture the CSM goes against the recommendation of *drawing the value stream map by hand and resisting the use of computers* [50].

5.3.3 Waste identification

In step S3, data available in the organization, or informed estimates are acquired for key parameters in the value stream, in particular, about the number of work units, artifacts produced, rework, estimates for the size and effort, defect data, typical processing times and variance are collected. To facilitate and to improve the reliability of results, the activity of identifying wastes should use a combination of qualitative (of data acquired in the workshops) and quantitative analysis (of empirical and simulation generated data). In particular, quantitative data is required to follow Strategy 3 for the use of simulation.

Out of the seven guidelines for lean value stream [50], we consider the following the most pertinent for software product development: *"continuous flow"* and *"pull"*. We briefly describe the two concepts below:

Continuous flow means that work items flow through the process at a continuous pace. A detailed account of definitions of continuous flow and reasons for the lack thereof is presented in [43, 42]. A lack of continuous flow, in combination with the waiting times, is a first indication of waste in the process, and should be further investigated [29]. An aim for lean is to ensure that one process produces only what and when the next process needs it, this will help achieve shorter lead-time and higher quality at a lower cost of development [50]. This has been illustrated in Figure 5.2. The figure shows that in the *"pull"* we observe large backlogs of work in all process activities, also referred to as partially done work [46]. Early in the process this is not an issue as requirements represent ideas that one can draw from, and no large investment has been made. Though, if the company has a large backlog of untested code, then an investment has been made in the implementation that does not flow back to the company in terms of revenue. Furthermore, when having too long waiting times, already implemented requirements may become obsolete [44].

Another supplementary concept is that of *"pull"* through which the process/phase, called the *"pace-maker"* (a phase with longer lead-time or with unreliable rates that determines the pace of the overall process), sets the pace for upstream processes. Thus,

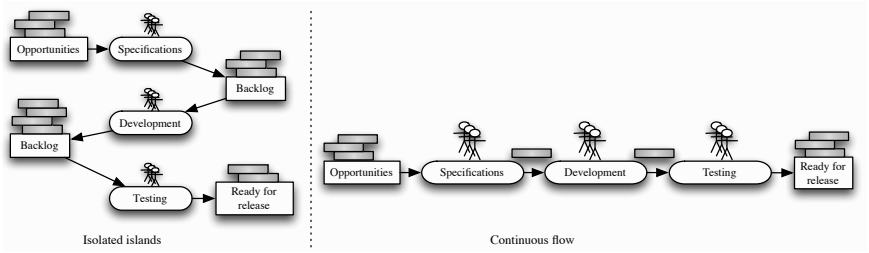


Figure 5.2: Continuous flow instead of isolated islands based on [50]

the work units are pulled by the *"pace-maker"* process instead of being pushed to it from upstream processes. In agile software development one may argue that the development phase is the *"pace-maker"*. Thus, the output of processes upstream (e.g. requirement specification) should be pulled by the development processes (e.g. design and development) and the downstream processes (e.g. verification, integration and release activities) should have a continuous flow.

Simulation assistance: Strategy 3 in Table 5.1 suggests the use of simulation to identify new bottlenecks or to verify the existing bottlenecks [2].

If the data to accurately simulate the entire CSM is available then an analytical comparison should be sufficient to verify the existence of bottlenecks and developing a simulation model just to verify known bottlenecks will be unnecessary. However, if there is a change that is going to happen in the future then simulation can be used to identify and verify any potential bottlenecks in the CSM. For example, if we know that in the future we will have to increase the number of maintenance requests we can simulate and see if the current process can handle the increased flow, and identify where the bottlenecks will appear (or verify if that concurs with our perception of bottlenecks) or are likely to manifest in the future.

In SPSM, almost any of the works on the use of simulation for strategic management [59], control and operational management ([37]), planning [3, 16], technology adoption and process improvement [17] can be used depending on the scope of the process and particular focus of change that needs to be investigated. A comprehensive coverage of industrial applications of SPSM for a variety of purposes can be found in Chapter 2.

5.3.4 Process improvement

In this step, we revisit the findings from the various analyses and discussions. In some cases, we also present results of lean transformation from other products within the company or from other companies having a similar industrial context. With these inspirations and points to consider, certain changes are proposed to eliminate waste and a future state map is drawn.

Typical questions for arriving at the FSM from the current state map in VSM literature are not entirely applicable in SE [50].

Simulation assistance: Given that quite often we will have more than one way of addressing a challenge in the process (or more than one way to eliminate a waste in the process), simulation is suggested to evaluate and choose from these alternatives. This is *"Strategy 4"* in Table 5.1 as depicted in the Figure 5.1. Thus, a decision can be made by evaluating the alternatives and then designing the FSM accordingly. Some literature on evaluation of improvement alternatives using SPSM include [17, 23], for a more thorough list see Chapter 2.

Khurum et al. [29] recommend to *"Re-evaluate value"* to assess the impact of changes that have been recommended to eliminate waste. The strategies for the use of simulation to assist in this step are Strategies 1 and 2 (as shown in Table 5.1).

5.3.5 Retrospective analysis

In this step, we collect feedback about the process and outcomes of VSM. This helps to identify shortcomings and improve future VSM undertakings.

In this study, Strategy 4 (see Table 5.1 and Section 5.3.4), that is to "develop a simulation model to evaluate several alternatives to derive a FSM" was evaluated using the research method described in Section 5.4.

5.4 Research methodology

According to Runeson and Höst's [51] adaptation of case study research, based on the purpose of the study, it can be classified as case study research with the aim of improvement. The phenomenon under study is *"value stream mapping"* in a software intensive product development context. We explored and evaluated the utility of simulation in supporting the value stream mapping.

5.4.1 Research questions

RQ: *How useful is the combination of simulation and value stream mapping, and what are its strengths and limitations in that regard in the software engineering context?* This question is answered through two case studies on two different telecommunication products at Ericsson AB. The following two sub-questions were addressed to assess the usefulness of VSM supported with simulation:

- *RQ1.1: Which wastes and improvements could be identified with VSM combined with simulation?*
- *RQ1.2: How do practitioners, facilitators and an external simulation expert perceive the process and outcomes for the VSM assisted with simulation?*

5.4.2 Research context

Ericsson AB is a large, ISO 9001:2000 certified telecommunication vendor. The two products studied here have different levels of maturity, type of users, and number of customers. Both products have different approaches to development, and are not part of the same overall system structure. Therefore, in this study they are considered two independent cases.

That is, their similarities are limited to shared aspects of company culture (e.g. terminology, etc.), the product domain, and the overall high level process, which are depicted in Figure 5.3. Therefore, in this study they are considered two independent cases.

The process starts by breaking down a *Business Opportunity* (BOP) into high-level *Business Use cases* (BUCs) that describe the system level functionality that the system must deliver to exploit the opportunity in the market. BUCs take into consideration the product anatomy, technical requirements and the business context and provide a more detailed specification. BUCs are the work-item used to plan, control and manage development progress in the company. A more detailed analysis of BUCs is performed and broken down to user-stories that are implemented by the system development teams.

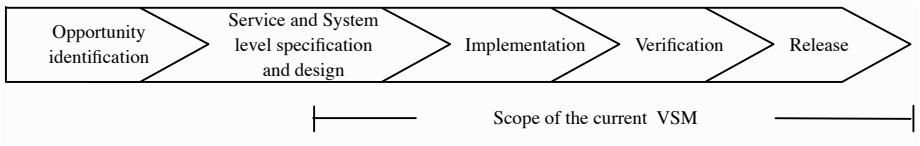


Figure 5.3: Overview of the product development process in the case company.

While both products use agile practices in development, there are structural differences in how the work is organized; e.g. one uses fixed-length sprints and the other uses a variable-length sprint (where the length depends on the estimated time required for the development of a BUC). For both products, a development sprint is followed by a three week testing-sprint that covers system-level testing. The two cases in this research are referred to as product *A* and *B*.

Product A: Product A is a relatively new offering from the case organization, but has a large potential market. Design and testing is done in 3-weeks sprints. The main programming language used is Java and GIT is used for source code version management. The following teams are involved in its development located in India and Sweden:

- One team each for system management, and system testing. System management is responsible to determine how to integrate the requirements in the form of BUCs into the existing system, as they know the system's anatomy very well.
- Seven teams for development (each with five developers and two testers)

Product B: It is a configuration product that has to interface with various databases and applications. It conceals the complexity of interconnectivity between different vendor software. It is not marketed as an independent product. It has one major release every year. Product development is done in two locations, China and Sweden, where the following teams are involved.

- One team each for system management, architecture support, test automation and system testing
- Four teams for development (each with 5 developers)

5.4.3 Data collection and analysis

In this section, different means of data collection used to answer the research questions RQ1.1 and RQ1.2 are discussed.

Documenting work products of VSM (RQ1.1)

We relied on the outcomes from the data analysis to reflect on the outcomes of the VSM, collected based on notes taken by the researchers during the activity. In order to increase the reliability of the data collected, one researcher took notes and the other researcher was mainly responsible for moderating the VSM activity. In addition, we took photographs of all collected materials on the white boards.

Reflection and observation of external researcher (RQ1.2)

To elicit feedback from the practitioners about the usefulness of both the process (that was followed to conduct simulation assisted VSM) and the outcome (of the VSM activity), we conducted a moderated workshop. Semi-structured questions and a structured questionnaire were used to systematically acquire qualitative feedback. Participants from both product A and B filled in the questionnaire (Table 5.2) anonymously. The coverage of various aspects of this research by questions posed in this questionnaire can be classified as shown in Table 5.2. The participants of the study filled in the questionnaire during a workshop session, which allowed them to ask questions for clarification. In addition, we acquired qualitative feedback through discussion during the workshop.

The same questionnaire along with additional questions about the benefit of the approach was used to assess the usefulness of the approach from the *"VSM Manager's"* perspective.

An external observer (the 3rd author) was present during the workshops. The external observer used four different approaches to acquire data: participation in the workshops as an observer; interviews with the VSM facilitators; document analysis and informal meetings.

Participation in workshops: The external observer during the workshops, took notes about observations related to the outcomes of the VSM activity, as well as what he observed in relation to the use of simulation from the perspective of a simulation expert.

Informal meetings: These unstructured/non-systematic meetings were important to capture explanations and increase the understanding of observed activities, roles, decisions, and contextual information, both from the case under study at Ericsson and the applied methodology.

Interviews: Two separate interviews with the researchers (first two authors of the study) involved in the case study were conducted, basically, to capture their opinions about how they perform simulation-related activities to support their SE empirical studies. The interview was semi-structured and the questions are available in [4]. The answers from the two researchers were used to compare and reflect on the perception of effort required by researchers in various steps.

As mainly qualitative data was recorded, the analysis of the observation records and notes was initially performed in isolation using open coding technique [48] to highlight and organize the main observed topics. Later, data collected from different sources were used together, making a triangulation to understand the reasons for the findings which are reported in Section 5.5.2.

Table 5.2: The aspects evaluated in the questionnaire.

ID	Question	*Evaluated aspect*			
		Process	Outcome	VSM	VSM + Simulation
Q1	'We have gained new insights about our software development processes which we did not have before.'		✓		✓
Q2	'Overall more insightful discussions resulted from the simulation results shown.'		✓		✓
Q3	'The simulation result shown was important to make a convincing point about the importance of reducing variance to achieve improvements.'		✓		✓
Q4	'The simulation result shown was important to make a convincing point about the importance of reducing waiting times to achieve improvements.'		✓		✓
Q5	'I would like to see simulation as an analytical tool for process improvement more intensively used in the company.'	✓			✓
Q6	'I do not prefer VSM to other software process improvement activities that I have participated in earlier.'	✓		✓	
Q7	'I would like to see VSM used more extensively in the company.'	✓		✓	
Q8	'The improvements chosen were influenced by the VSM activity.'		✓	✓	
Q9	'The improvements identified are realistic to implement.'		✓	✓	
Q10	'The improvements identified will help in improving the software quality.'		✓	✓	
Q11	'The analytical phase (current map) did not provide valuable input to design with the future state map.'	✓		✓	
Q12	'The results of the empirical study [54] influenced the choice of improvement actions.'	✓		✓	
Q13	'The improvement actions identified will be implemented in the future.'		✓	✓	

5.4.4 Validity threats

Petersen and Gencel [41] reviewed the literature on validity threats based on different scientific schools of thought and recommend a categorization of validity threats encompassing different views. They recommend discussing 1) descriptive validity (factual

accuracy), 2) theoretical validity, 3) generalizability (internal and external), 4) interpretive validity, and 5) repeatability.

Descriptive validity is concerned with describing observations made objectively and truthfully.

- There is a threat that the researchers captured information and views provided during the workshops inaccurately. Two actions have been taken to reduce this threat. First, while one of the researchers was having the main moderator role, the remaining researchers were recording the information provided by the participants taking notes. Hence, the data collection did not rely on one individual. Second, the findings and interpretations of the researchers were presented at the beginning of each workshop (member checking).
- We utilized a questionnaire to capture the views of the practitioners, a threat being that questions might be misunderstood. To reduce this threat, the researchers were present while the practitioners filled in the information, so that they could answer questions for clarification.
- The external observer who interviewed the first two authors of the study, and also acted as an observer during the workshops, was new to the company and their terminology, while the first two authors had some familiarity with the domain and the company.

Theoretical validity is concerned with our ability to capture what we intend to capture, the major threat being confounding factors we are not aware of. It may be affected by the trust of the practitioners towards the researchers. To reduce the threat, the VSM manager was present during the workshops. Furthermore, the first two authors have a history of working with the company and could refer to previously conducted projects to highlight the trust. Hence, the threat is reduced, though not mitigated.

- The selection of subjects largely influences the findings. The key stakeholders important for the activity have been identified with the help of the VSM manager. The VSM manager assured that the persons in the workshops have excellent knowledge of the process, and could also assure that actions identified in the workshops are actually implemented.
- Expectations and attitudes towards the VSM activity may affect the results, and hence may alter the results. Overall, given that the practitioners to the largest part evaluated the activity positively, this is not considered a significant threat.

Generalizability is concerned with transferring the results within a context (in this case within the cases) and between different contexts (e.g. other products and companies).

- To assure internal generalizability the VSM manager supported us in the selection of workshop participants. All activities of the overall process were covered, which reduces a threat to internal generalizability.
- We conducted the study in two different cases that were developed at the same company, but followed different processes and both products were of different ages. Product A was a newly developed product, while product B was mature. For product A and B the baseline situation was very different in terms of availability of data, which had a significant effect on the ability to use simulation. Furthermore, contextual factors may exist that may affect the process of combining simulation with VSM that could not be captured in this study. In order to facilitate the judgment of the generalizability, we provided a detailed account of the context, and the background of the practitioners.

Interpretive validity is concerned with whether the conclusions and inferences made by the researchers based on the data are reasonable or not.

- All three authors have worked with the creation of simulation guidelines ([12] and Chapter 4), and have written synthesis studies on process simulation ([13] and Chapter 2). The conclusions drawn based on their previous work may affect their interpretation of the findings. In order to reduce this threat, we investigated the results from (a) the views of the practitioners; (b) the researchers conducting the simulation study through the interview conducted by an external observer (third author), and (c) the external researcher acting as an observer during the workshop. These findings could be triangulated. Hence, the threat to interpretive validity is reduced.

Repeatability is supported by describing the research method in detail. Furthermore, we provided a web-page containing the instruments for data collection and the non-calibrated simulation model as the raw data could not be made available due to confidentiality reason.

5.5 Results of the evaluation (Contribution 2)

This section reports the second main contribution of the study that focuses on evaluation of simulation assisted VSM. In this regard, the proposed framework (presented in Section 5.3) was instantiated at the company and the results are presented in Section 5.5.1, answering RQ1.1 concerned with the wastes and improvements that could be identified. Section 5.5.2 answers RQ1.2, which is concerned with the perception of the participants in the VSM activity.

5.5.1 Conduct and outcome of the VSM activity at the company (RQ1.1)

The VSM started with assignment of the following roles and responsibilities to participants of this study:

VSM Manager: In this study, one practitioner took on this role, he had the authority and the power to facilitate and ensure that change happens in the organization. He was also instrumental for conveying the goal and convincing the teams why it was important. He helped in negotiating the appropriate scope and goal for each product. He also ensured that the necessary resources and time was allocated as a priority for VSM workshops.

VSM Facilitators: The first two authors acted as VSM facilitators in this study. Both authors have experiences as facilitators, and the second author has conducted VSM studies earlier.

VSM Participants: In large product development, as is the case in the products being studied here, no one individual will know all the details of the process. Besides, they may feel more strongly about the challenges they face in the work that they are more closely involved with, this may lead to sub-optimization which is contrary to the aim of VSM. There were 5-6 practitioners from each product, with knowledge and experience encompassing the end-to-end process ensuring a variety of perspectives. The number of participants and their median experience in years with the product and the case company is summarized in Table 5.3. These participants covered all the major phases starting from system management all the way downstream to system testing.

Initiation: Goal for the VSM activity by the upper management was stated as the following: *Reduce the average and variance in end-to-end lead-time by half of its current value without compromising the quality.* Three target aspects highlighted here were the efficiency (measured by average lead-times), predictability (measured by variation in lead-times) and quality (the number of defects detected both internally and by external customers).

For the two products, it was translated as *"Reduce the lead-time (on average) from "Start BUC analysis" step to "BUC is ready for release" step by "50%" from currently*

Table 5.3: Participants of the VSM workshops

Product	Number of participants	Median experience with the product (in years)	Median experience with the company (in years)
A	5	1.5	11.5
B	6	5.5	18

X "units of time" and the variation in lead-times by "50%" without compromising the quality." For the sake of confidentiality we have not reported the actual number or units for the current lead-time.

Once the goal was agreed upon, the product owner made available to us process documentation and measurement data on end-to-end lead-times and any sub-divisions of lead-times that were available for the sub-phases. To an extent, this helped the facilitators get familiarized with the overall product development process.

Current process map: This workshop is the first interaction with the participants from the team. So the *"VSM facilitators"* started with introductions followed by what is VSM, how it is done, what are the benefits and what is expected of the participants in the rest of the workshops. The *"VSM Manager"* then presented the goal and the motivation behind it.

Then the participants were asked to *"take one typical BUC through the process from the first to the last step as identified in the goal"*. This way we were able to elicit the information expected in a CSM without forcing a certain notation on the participants. This understanding of the current way of working was fundamental for the next step to identify and understand the "waste" in the process.

Waste identification:

- **Contradiction between perceived challenges and data:** This was observed when the perceived challenges were compared to the quantitative data analysis results that are typical of VSM for identifying *"wastes"* [40] in the process.
- **Difficulty to estimate process metrics in the absence of data:** Experts find it hard to quantify waiting times and variations in the development process and any critical reflection based on calculations or simulation from such estimates is disregarded as inaccurate.
- **Contradiction between the expected and likely benefits of improvements:** This was possible when proposed improvements were assessed using simulated output with explicit assumptions.
- **General applicability of data analysis and visualizations:** This became evident from the positive feedback we received during the workshops and how these reports are part of the corporate measurement dashboard.

The remainder of this section details how these findings were derived from the data.

Contradiction between perceived top challenges and data

After reaching a consensus on the accuracy of the CSM, the participants were asked to identify key challenges in achieving the stated goal.

Once a list of challenges was identified, practitioners prioritized them in terms of their potential to contribute towards the stated goal, provided that they are overcome or ameliorated. The top three challenges identified for the products in this study are presented in Table 5.4. A rudimentary root-cause analysis was performed to understand the causes for these challenges.

Table 5.4: Challenges and prioritization for the two products.

Product A		Product B	
Challenges	Votes	Challenges	Votes
Code freeze at merge, since all development sprints are in sync	17	Time spent on initial analysis of BUCs	16
3-week sprints (will not complete something 1.5 weeks earlier Parkinsons law)	16	Large backlog of BUCs waiting to start	16
Quality of BUCs	15	Testing-sprint length	6

At this stage, it is interesting to consider that only the expert opinion was captured. Next we did analysis to determine whether the perceived challenges can be backed up with quantitative data.

Lead-times: By clustering the BUCs lead-time by their estimated effort we could identify BUCs that were the best or the worst in their category in terms of how long they took to go through the process. The lead-time of BUCs was grouped according to the estimated effort using different units of measure as used by the two products. The outcome of this analysis for both the products is shown in Figure 5.4.

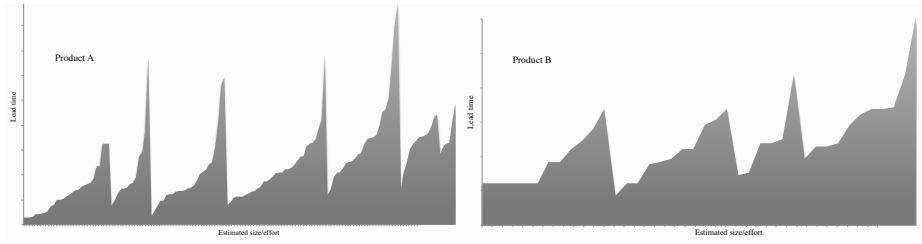


Figure 5.4: End-to-end lead-time for BUCs grouped by their estimated effort.

Figure 5.4 shows that the variance in lead-times is omnipresent throughout the size clusters i.e. certain BUCs with similar estimated time for development, take considerably longer (and hence the peaks).

The challenges identified in the first workshop were general (see Table 5.4) and should affect development times of all BUCs and hence do not explain why a small percentage of BUC (albeit of different sizes) take so much longer than the average case (see Figure 5.4). Excluding the lead-times of BUCs that took longer than *"X"* units of time, which means about 30 % of the BUCs, the average lead-time for the remaining BUCs was very close to the target of reducing lead-time by 50%. One hypothesis could be that by finding out the reasons why these 30% of the test cases take longer, and alleviating the reasons behind it we can achieve the target lead-time. And such potential improvement was not expected from addressing the top challenge the experts perceived as major hindrance to reaching the target lead-time.

Waiting times: These are another way to identify inefficiency in a process by analyzing the time that a work-unit is inactive (referred to as waiting time) [34]. The flow of BUCs through the process was analyzed by distinguishing between waiting times and productive times. This was used as input for reflection on the duration of waiting times, the reasons behind them, whether these were acceptable delays, and if not how can these be reduced.

For product A, the waiting times are shown in Figure 5.5. The data to do this analysis was not available for all the process steps in Product B, however, it was noticed that there is a long waiting time at the start of the development process where an initial analysis has already been done for work that will was not going to start in the near future.

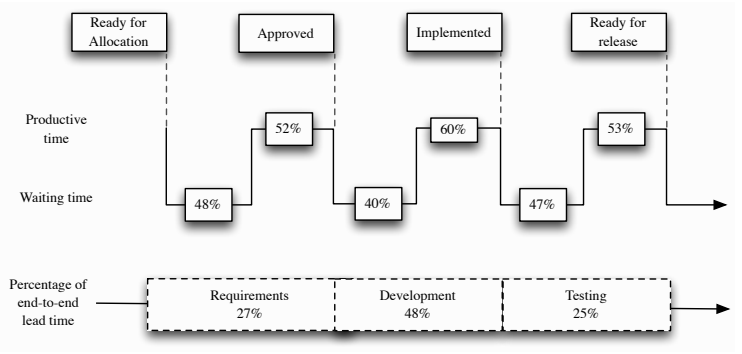


Figure 5.5: Waiting and productive times for Product A

For product A, as shown in Figure 5.5 almost half of the time, work packages are waiting in backlogs. It also highlights that the artifacts that are idling are becoming obsolete e.g. the preliminary analysis done in case of product B will need to be redone if development starts after a long time. Besides, another threat is that it might not even be relevant by that time. These waiting times within the process also highlight bottlenecks and where the flow is breaking up in the process.

It was again seen that the challenges identified through expert opinion (see Table 5.4) could not explain the long waiting times in the process. Furthermore, the challenges identified did not have as significant improvement potential as tackling the waiting times offers. Besides, improving the productive time is perhaps much more difficult compared to reducing the waiting times to achieve the specified targets.

Visualizing flow: Waiting times are often correlated with a discontinuous flow of work. Different ways of visualizing the flow are used at the company, e.g. cumulative flow diagrams [43]. In this study, we also used time series analysis (see Figure 5.6). Figure 5.6 shows the number of BUCs ready for specific activities (e.g. specification in further detail, or development, etc.). As can be seen in the figure, discontinuity is visible in BUCs *"ready for testing"* and BUCs *"ready for release"*. That is, during a high number of weeks no BUCs are ready. Furthermore, in some weeks a very high number is ready, while in other weeks only one BUC is ready.

The visualization of flow in Figure 5.6 made it apparent that the perceived improvement to reduce the lead-time by having continuous testing cannot be justified unless the variability in development process is improved. Thus, the additional cost of shortening the test iteration, that requires additional cost in terms of test automation and acquiring more resources for testing, cannot be justified in the existing development flow.

Flow analysis was only conducted for Product A, as Product B did not collect the required data. This also had important implications for the use of simulation. That is, only a simulation model was built based on the data acquired for Product A.

Difficulty to estimate process metrics in the absence of data:

Poppendieck and Poppendieck [46] recommend that by working with people involved in various activities in the development process one should get the average amount of time spent in each activity and also the breakdown of the time into waiting and non-value adding states. In our experience, although we had a sample of representatives for each activity in the process, the participants could estimate the average time it takes for each activity, but they found it impossible to identify how long does an average use case spends in a waiting state. Therefore, this analysis can only be done and have meaningful results if there is an observed dataset to supplement it, as was the case for Product A. In cases of product B the waiting times were not collected by the organiza-

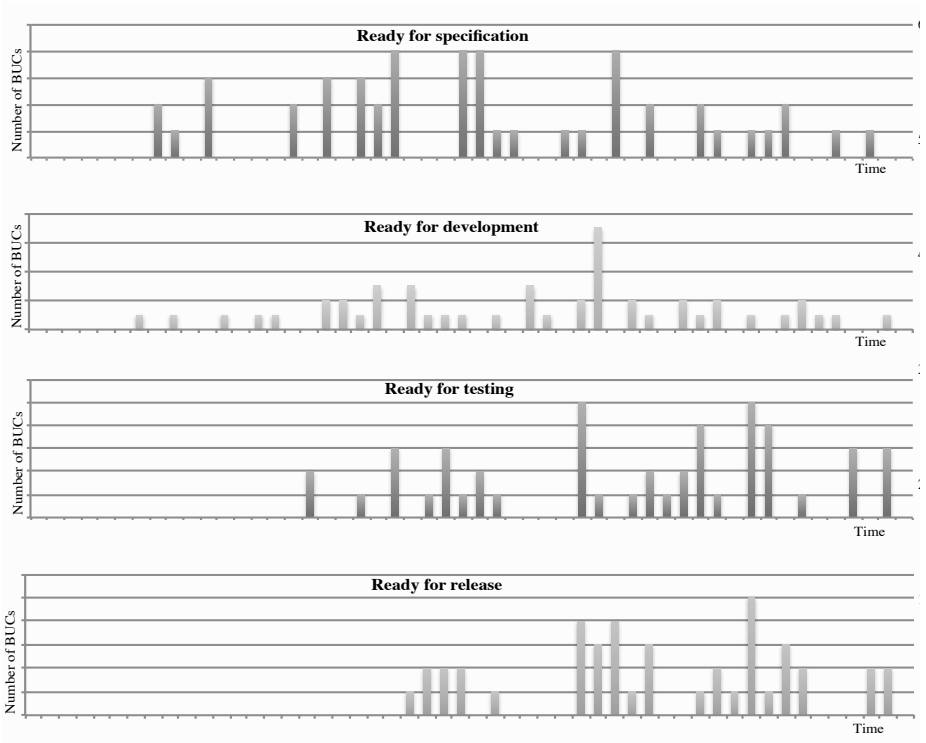


Figure 5.6: Assessing the flow using a time series plot

tion and the participants found it very difficult to estimate general rates. Furthermore, an analysis on top of these estimates was not considered reliable as this was all based on *"guess work"*.

Some additional interesting observations were not possible using the simulation approach, such as causal influences among variables and the process activities. The main reason are associated to the available data. Since the organization had no goals for simulation before, no data was available for that. This way, it is not possible to discuss reasons explaining the results due to the lack of a mechanism.

Contradiction between the expected and likely benefits of improvements:

Simulation based analysis: In this study we employed Strategy 4 (see Table 5.1) to use simulation to assess various alternatives for improvement to derive a future state map. Simulation was used to reason about the likely implications of transitioning towards continuous flow (see Figure 5.2) instead of the current fixed three weeks iterations. The current way of working (Case 0) and the intended way of working (Case 1) are depicted in Figure 5.7, which were assessed using simulation. A static representation of the simulation model can be found in [4].The aim with the intended change (from Case 0 to Case 1) was to reduce lead time, improve quality through early feedback, and to develop the capability of shorter release cycles for the future.

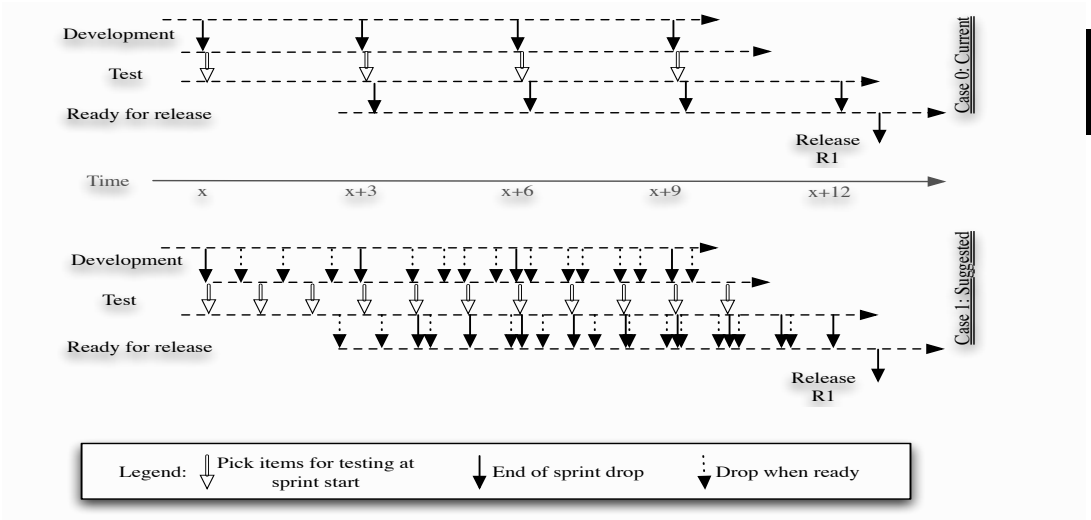


Figure 5.7: Current development process and suggested development process.

Case 0: Three weeks sprint for development, three week test sprints with hand-offs at the end of the sprint for both development and testing.

Case 1: Three weeks sprint for development, one week test sprint with continuous drops from both development and test.

For stochastic simulation, an important step is to identify the random distribution that accurately represents the underlying process behavior. Empirical data from the product was available for the last calendar year, however, the team considered only the data from the last six months as representative of the current process as it had undergone change around that time. This data unfortunately was not sufficient to identify

a probability distribution with any statistical confidence. Therefore, for arrival rates of new BUCs, the actual data was used and for development and testing rates normal distribution was used with mean, minimum, maximum and standard deviation as observed in the real data. The reference behavior of the model was checked by determining whether the total cumulative number of BUCs is similar between the simulated case and the actual case, which was true.

Scenarios: Three key scenarios were executed with the help of the simulation model. *Scenario 1* shows results of the two cases with the parameters based on current data. *Scenario 2* shows the outcome of the two cases if the productivity of test and development team stay the same but the variation is reduced (a more predictable process). *Scenario 3* shows the outcome of the two cases above if the waiting times in the process are reduced. The results of the simulation model in terms of cumulative number of BUCs that are ready to be released for the above scenarios can be seen in Figure 5.8.

From the simulation results (in Figure 5.8) it was illustrated that the benefits of changing to a shorter test iteration are not very significant. This is because based on the current data there is no continuous delivery from development that a shorter test iteration can leverage from. The result of *Scenario 2* further shows that reducing the variance in the process will lead to more significant gains than just the shorter test iteration alone. Lastly, *Scenario 3* shows that an even greater potential exists in reducing the waiting times in the process.

It was identified that often the developers have a tendency to wait for the deadline to commit changes and to update the BUC status. From the discussions based on Scenario 1, the importance was again highlighted for accurate measurements to predict the likely outcome of change.

200 replications of Scenario 1 were done and the outcome was plotted. The variation in the outcome highlighted the unpredictability of the process and the lack of confidence with which we can forecast the impact of the suggested process change. This along with the results of Scenario 2 highlighted the importance of dealing with the variance. Roother and Shook [50] also noted that if the work occurs unevenly overtime (in the form of peaks and valleys) it results in extra burden on people.

Thus, the simulation results showed that the challenges that practitioners perceived as the main impediments to achieving the target of reducing lead-times for product development do not explain the major waiting time and variation in the process data. Furthermore, simulation results showed that the benefits expected from addressing the perceived challenges are highly unlikely to deliver the promised results (e.g. due to the current flow of BUCs from development to testing, continuous testing can only be leveraged upon if there is a continuous delivery from the development teams). Furthermore, the high variation in the flow data, became visible in the unpredictability

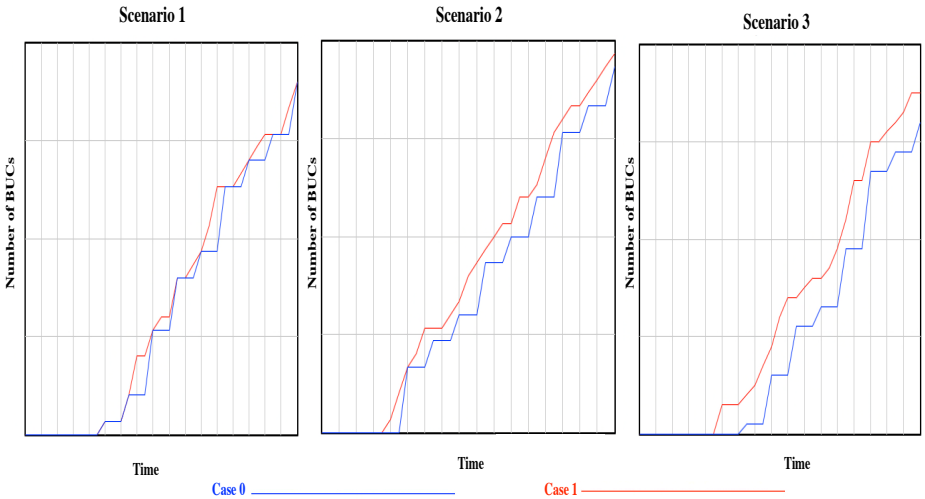


Figure 5.8: Three scenarios of comparing the current way of working with the suggested changes.

of simulation results thus again emphasizing the importance of more significant challenges that exist in the process and have not been identified by the practitioners.

General applicability of data analysis and visualizations:

Figure 5.4 showed similar patterns for both the products although the units for estimation were different. This points to a general applicability of this analysis in the case company. Also, the list of BUCs identified this way served as a good basis for reflection and validating the preconceived challenges that the team had about their process. The company will soon have it implemented in the dashboard. Already the list of top ten BUCs with the longest lead-times foe end-to-end and per phase are available in the dashboard.

The time series analysis (as shown in Figure 5.6) was found more intuitive by many practitioners to visualize the flow of BUCs and has also become part of the dashboard for the company.

In the case company, for both Product A and B, the utility of tracking the waiting times also became evident. For Product A, where the measurement points were already in use, all the statistical and simulation based based analysis was possible. While for Product B and others, there is now a company wide initiate to keep track of some

"mandatory" and some recommended measurement points that a team has to collect to keep track of their development process.

Process improvement: In the penultimate workshop, the key findings from all the analysis and reflections in the previous workshops were revisited. For product B, case study results from another Telecommunication company that has transformed from agile to Kanban approach [54] were presented for inspiration. Teams developed an action plan with a list of changes and improvements and drew the FSM.

5.5.2 Evaluation of simulation assisted VSM from various perspectives (RQ1.2)

This section reports the results of evaluating both the process and the outcome of conducting simulation assisted VSM from different key perspectives.

Practitioner's perspective

Usefulness from VSM Manager's point of view: Questionnaire (see Table 5.2) was used to elicit VSM Managers opinion about the process and outcome of the simulation assisted value stream mapping. Compared to participants feedback in Figure 5.9, except Q12, where VSM Manager considered that the results and lessons from a published study did not have influence on the choice of improvement actions. On Q1, Q5 and Q7 he indicated strong agreement and on Q2, Q3, Q4 and Q8 he indicated *"moderate agreement"*. He *"strongly disagreed"* to the statement Q6 that he does not prefer VSM to other software process improvement. This clearly shows favorable feedback on outcomes of VSM, use of simulation, the process of conducting simulation assisted VSM.

Usefulness from VSM participants' perspective: The responses of nine practitioners from the two products using a Likert scale are depicted in Figure 5.9. Please note that Q6 and Q11 were negative questions (therefore semantically it shows positive feedback for VSM process activities for using CSM and VSM in general). The only questions that got negative feedback were Q3 and Q4, where 22% and 20% of the respondents, respectively, were of the opinion that the simulation results were unable to make a convincing case for targeting the *"variance" and "waiting time"* in the process for improvement. Overall, for questions we got positive feedback with some *"strong agreements"*, especially on Q9 and Q10 where 50% of the respondents strongly agreed on the realism and likely impact on quality of the improvements and the other 50% moderately agreed to it.

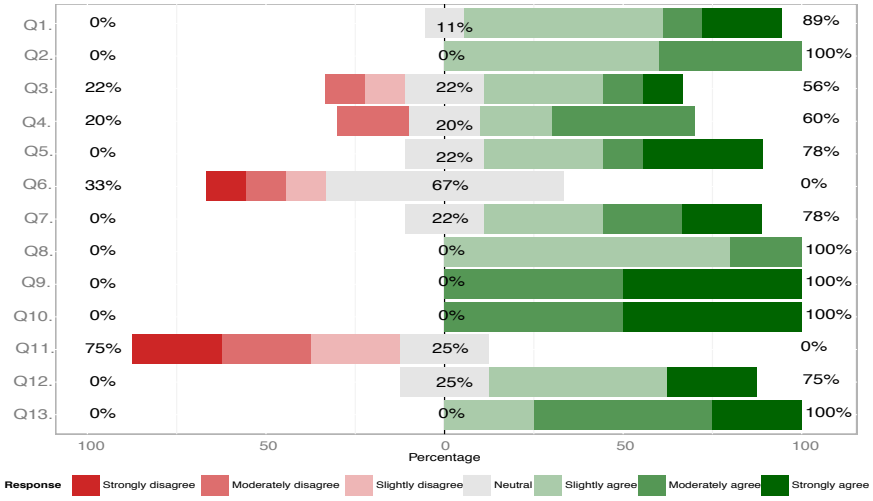


Figure 5.9: Results of the questionnaire (see Table 5.2 for questions) to evaluate the simulation assisted VSM.

External observer

The external observer noted down his reflections by observing the activities and outcomes of the workshops, and by interviewing the VSM facilitators. His key findings are described below:

Suitability of employed simulation approach: Based on the information and documents given (process descriptions and workshop records), it is possible to observe that problems regarding the product A investigated with simulation resembles a process perspective, evidenced by some characteristics such as synchronization of parallel flows and analysis of waiting times. Probably, it would be easier explored using a discrete-event simulation model, even though such problems can also be handled in system dynamics.

Feasibility of simulation depends on data availability: Although there are VSM-related reasons for adopting simulation as an investigation approach, the data availability was crucial for defining the simulation feasibility. Otherwise, such approach would be prohibitive.

Utility of simulation assisted VSM: The VSM workshops create a worthy environment for discussing challenges and alternative solutions, from which scenarios (for analysis using simulation) can be determined. These workshops can be potentially used

as room for discussing the simulation model, acquire input to determine relevance of variables to reach consensus about what is being simulated, and to validate the simulation model.

When more variables of interest or more scenarios appear during the interactions with the development team, the analysis procedure for simulation outputs will certainly change and needs to be revisited. Therefore, researchers will need to iteratively discuss the results with the team after each revision.

Process for simulation model development and lack of validation: The planning steps for the framework, concentrate on the VSM activities (see Section 5.3). However, there are no simulation-specific activities or steps in it. Even using the previously discussed strategies (see Chapter 4), important activities of simulation-based studies are not explicit, such as the simulation model assessment and experimental design, and may have been overlooked by the researchers.

The adopted modeling approach in the study followed an iterative process [6]. During the observations, the external researcher could not identify any attempt from the researchers to validate the simulation model with the practitioners.

For the model assessment, the only observation made regards the constant review of the simulation model, when unexpected results started to appear, i.e., when simulation results differ from the actual process behavior. Initially, it appears to be a consequence of the mismatch between the simulation approach (SD) and the problem under investigation, but it also seems to be the selected procedure for model verification.

Besides the resistance shown by one of the teams regarding the statistical analysis of the process metrics (collected in the organization), there was no questioning about the simulation model validity. Apparently, they assumed the simulation model was correct or, maybe, became overwhelmed by its complexity.

When trying to support VSM by complementing with simulation, the researcher must be aware of the new threats simulation imposes. Even when not having prediction as a goal for the simulation model, its use can still be planned.

Simulation models were not presented upfront: It is important to highlight that, for the two first workshops involving both product A and B, the researchers decided not to present the simulation model. The discussions were based only on the presented simulation results and statistical analysis of data. Only at the third workshop, the researchers presented the simulation model and outputs from the experiments involving the different scenarios. At first, stakeholders appeared a little confused on how to read the model, maybe because of the system dynamics notation. Later, some stakeholders apparently changed their opinion with the simulation tool and its possibility of anticipating scenarios.

Difference in perceived effort spent on simulation modeling: In the interview conducted at the end of the period of observations, it was possible to identify distinct

perceptions of effort spent on simulation issues. The researchers have different opinion on which activities they spent more effort: input data analysis or simulation model development. The input data analysis involve the understanding of the company data, using statistic techniques in order to identify or explain certain behaviors. On the other side, the modeling is more related to how to translate the process specifications (CSM or FSM), along with project data, into the System Dynamic model, which showed to be problematic at the beginning by the mismatch between the SD approach and the kind of problem to be simulated. Such different perceptions might be just a matter of roles played on the study or different kind and level of expertise.

5.5.3 Retrospective after six months

In October 2014, we returned to the company six months after the conclusion of the VSM activities. This gave us an opportunity to capture information about the impact of the VSM activities on the organization. Two follow-up workshops were carried out to assess the status of the two products in implementing the changes and the adoption of VSM in the company.

The two teams presented their status on each action item and a comparison was done of lead-times, size of backlogs and other measures that were used in the earlier VSM activity. Overall, teams had made significant progress towards implementing the improvements identified through the VSM. At the time of these follow-up workshops there were only slight improvements in the lead-time and variations in the process.

In terms of adoption of VSM, it was found that some reports for doing data analysis, as done in the two case studies reported in the chapter, have already been implemented in the corporate dashboards, and remaining will be made available in a later release. It is well aligned with the advice in software process improvement literature to ensure automatic collection of data [22].

So far, the practitioner who took on the role of VSM Manager in this study has conducted VSM for six other products. He stated that he received positive feedback from the teams that were involved in VSM. They considered that the benefits of the workshops outweigh the time spent by participating in them. Even with product teams with skeptical participants the value became apparent once they had participated in the workshops. The dashboard based reports were also useful to reflect objectively on the perceptions that participants had about the state of their product development (e.g. size and place of backlogs). However, the dashboard is not currently used as part of routine work and they will have local *"VSM facilitators"* to get VSM rolling in the organization.

He considers that the feedback on incorporating simulation has been positive, but there is a need to make it more accessible to practitioners that will take on the role

of *"VSM facilitators"* in the organization. Given the consistent one process that is followed through-out the organization and the standard suite of measurements that are collected in each of the products provides some avenues for the use of simulation. The expectation is to explore further the possibility to train the *"VSM facilitators"* to be able to use the simulation models even though they will not be required to create the models themselves. This would provide them with the tool-set to perform certain analyses and highlight the importance of having continuous/even flow over the typical resource utilization mindset.

5.6 Discussion

5.6.1 Evaluating the use of simulation to assist VSM

From the results in Figure 5.9, to question Q5, 78% of the respondents had a favorable opinion on use of simulation for SPI activities in the company. Furthermore, all respondents agreed (Q2) that the use of simulation led to more insightful discussions. This reflects positively on the use of simulation to assist VSM.

However, when asked about the two main points that were emphasized through the use of simulation, the importance of reducing *"waiting times"* and *"variation"* in the process, with regards to the *"waiting times"*, in Q4, 60% agreed that simulation helped to make a convincing argument. While on the second point of targeting *"variations"* in the process, only 56% agreed that the simulation results made a convincing point. Over 40% of the respondents did not agree or were indifferent on the impact of simulation for the two points in the discussion. These respondents were either still unconvinced about the importance of *"reducing variance"* as a target or they approached the question differently like *"simulation was not the sole contributor"* in making a convincing point.

The fact that simulation results led to more insightful discussions is enough to justify its use, since the purpose of employing simulation here was to support in decision making and not to make the decisions. However, the cost and time need has to be considered on a case-to-case basis. For this study we approximately (this is a rough estimate as it was done along with other work for VSM between the workshops) spent 10 calendar days on eliciting the scenarios, development and testing of the model, calibration with the data and to analyze and create reports based on the results.

5.6.2 External observer's critique on the validation of simulation models

The third author observed that not enough effort was spent on validation of the model. We utilized the simulation to evaluate alternative scenarios on how to construct the FSM. In Chapter 4, we proposed guidelines on how to conduct software process simulation. As part of the process, verification and validation of the model play an important role. However, the degree of verification and validation to be conducted may depend on the purpose of the model usage. In case the model should be used for predictive purposes, and shall be continuously used, a detailed verification and validation should be conducted (including sensitivity analysis, model walk-through, and reference behavior checks).

In this case, the model was checked for correctness by using non random input and analytically calculating the output at each step and the total output from the simulation model. In this study, the simulation model had to capture two scenarios (as shown in Figure 5.8). First author developed and tested the model, both the second and third author reviewed it and a walk-through of the model was given to the practitioners. In the walk-through we explained how the model represents the two scenarios.

Although data for one calendar year was available, however, the practitioners considered the data from the last six months as representative of the current process. This left us with insufficient data points to identify a random distribution that can represent the data with statistical confidence. We resorted to using random distribution with minimum, maximum, median and standard deviation based on the actual data. Due to this reason comparison to actual process behavior was not performed. Sensitivity analysis to find critical factors in the process was not done however 200 replications for each scenario were performed and cumulative output was plotted to visualize the dispersion of results.

The intention in this study was to design a *"throw-away"* simulation to quickly play through alternative scenarios, which meant that only the rough patterns (e.g. there is an effect of waiting time on the possible velocity of the organization) were of interest, rather than saying what the exact gain of velocity is. Similarly, the effect of variations in the process was made visible by the unpredictability of the simulation results in both scenarios when it was run multiple times.

Therefore, we consider that under the time constraints of the workshops and primarily due to the purpose of the model the validation done in the study was sufficient for the task. As the goal here is not to develop a perfect simulation model but to have a model that makes the assumptions visible and explicit. It will also assist logical reasoning and argumentation for why a certain alternative is better than others.

5.6.3 Use of VSM over other SPI activities:

In response to Q6, only 33% of the respondents favor VSM over other activities and a majority of 67% were neutral. However, a positive sign was that none of the respondents indicated that they will not prefer VSM over other SPI. The large majority of 67% could be because the practitioners did not have a comparison point (i.e. they have not participated in any other formal SPI activity before), or in the worst case they are weary of the SPI activities and see this *"as good as any other improvement activity"* they have participated in. Unfortunately, we do not have the information to ascertain if the respondents have an experience with other SPI activities. Thus, we cannot interpret the *"neutrality"* of responses here.

33% favorable responses suggest some inclination towards VSM as a SPI activity of choice. Furthermore, responses to Q7 where 78% agreed that they would like to see more extensive use of VSM in the company, supplements this conclusion. The overwhelming positive feedback on the outcome of the VSM activity in the following questions also adds to the confidence in the utility and encourages its further adoption in the software product development.

- 89% of the respondents (in Q1) agreed that they gained new insights about their development processes that they did not have before.
- 100% positive feedback on VSM leading to the identified improvement, the potential to improve the software quality, and that the improvements are realistic and will be implemented in the company (see Q8, Q9, Q10 and Q13).

The second point above also highlights another strength of VSM. Although, the initial drive for improvement came from the top management, but the participants considered the improvements realistic, with the potential to improve quality and their commitment that they will be implemented. This indicates the ability of the VSM activity to create the *"commitment"* which is crucial to the success of any SPI activity [22]. The participants get the end-to-end perspective and they can understand why a certain change is necessary in their process, although they may not be the immediate beneficiaries of the change.

5.6.4 Other process related considerations:

Application of VSM in SE is relatively new, but the lessons from research on success-factors for software process improvement initiatives are equally applicable in this context. For example: having realistic goals, executive support, sufficient resources, experienced and respected staff, and involving developers as well as managers in the improvement initiative [20].

Most of these factors are inherently encouraged in the bottom-up, non-prescriptive process of VSM, however awareness and conscious consideration of these during the VSM will further improve the likelihood of success.

The role of "VSM Manager"

The importance of having *"change agents"* is well established in software process improvement research [20]. In this study, he was critical to *"negotiate"* a goal for the VSM activity with the product owners, that is both relevant and acceptable (based on the target from the top management). He was also instrumental in ensuring the resources that were required for the activity, which primarily involved practitioners' availability for the workshops and access to data for the researchers (who took on the role of VSM facilitators).

Terminology

In the initiation meetings before the start of actual VSM workshops, we noticed some reservations on the use of the terms *"value"* and *"waste"* where some product owners considered it too academic. Therefore, in our communications and in the workshops the emphasis was rather on identifying challenges/impediments in achieving the stated goal, or on identifying the activities and enablers that can help achieve the stated target. Through this wordplay, we managed to avoid a lot of potentially unnecessary discussions without compromising on the aims of VSM where the focus was to improve the end-to-end lead time of the process.

Notation to draw CSM

Contrary to previous experience [29], the respondents in the current study (see Q11 in Figure 5.9) considered the CSM as useful input in identifying improvement and defining the FSM. This was achieved by not limiting the participants to use any particular notation, which in the previous case was considered too restrictive and simple to capture the actual process [29]. In our experience, giving practitioners the freedom to choose the notation at their will, helped us to focus on what is important i.e. drawing and developing a consensus on how they currently work.

Time taken by VSM

In the guidelines for doing VSM in the manufacturing domain, Rother and Shook [50] recommend that VSM should be a short non-bureaucratic activity where in two days

one should have a FSM drawn and ready for implementation, which one may *"fine-tune"* as the progress is made on implementation. However, in this study, it took considerably longer in terms of calendar time to arrive at the FSM, which was due to scheduling challenges from busy schedules for both practitioners and researchers.

Although, the two researchers have a background working with the case company, these products were new for them and it took some time to familiarize with the technical jargon. Also, all the challenges related to data access, extraction and understanding took some time as well owing to dependence on the industry practitioners. Lastly, all the analysis was done manually using spreadsheet and statistics software which required extra processing to clean up the data. The actual workshop time, however, for each product was approx. around 15 hours.

Now that we have practitioners from the case company doing VSMs internally, the challenges that we faced as outsiders will be redundant (the downside however will be the lack of a totally fresh perspective). Furthermore, the reports and visualization that were generated manually to assist in these workshops are now created automatically through a browser-based dashboard therefore that time will also be saved. Theoretically, it may be feasible to do these workshops in two full days now, but we consider that the workshops should at least be spread on three calendar days. As the workshops were fairly intensive, both for the participants and the facilitators, as often you will have follow-up questions that will require some additional data collection and analysis to answer them.

This will however, still fall well short of the recommended *"half hour"* threshold for VSM activity set by Kent Beck as quoted by Poppendieck and Poppendieck [46] *"it took us about half an hour to come up with this (don't spend longer than that or you will have too much detail)"*.

Waiting times

Interestingly, almost 50% waiting time found in the product development processes here (see Figure 5.5) is very similar to the typical waiting times found in engineering and manufacturing processes (where work packages were found to be inactive 60% of the times) [34]. This shows the usefulness and applicability of VSM for software product development as well.

5.6.5 Adoption of the proposed analyzes and the VSM process guidelines at the case company

We have been involved in two subsequent cases of VSM at the case company as well. In one of these, a practitioner from the case company (who was involved in the VSM

for product A and B) shared the VSM facilitator role and in the latest instance, she took over the entire responsibility and the researchers were involved in a supportive role (doing the data analysis in the background). The practitioner is also involved in training other employees in sites in Germany, China, India and the US to facilitate local teams to start adopting VSM at their end-to-end processes.

As mentioned in the previous section, all the reports of quantitative data analysis on lead times and flow are now part of the VSM dashboard to facilitate the analysis of their development processes. There is also an active education drive explaining and motivating teams to report the new measurements, use the dashboards and conduct VSM.

5.7 Conclusion

This study reports the strategies, the process for assisting VSM with simulation, and the results of its application in two industrial cases. Of all the strategies identified for the use of simulation to assist in VSM (see Section 5.3). In our experience the least likely to succeed for software product development is Strategy 5, although this is the one with the most theoretical potential. Except Strategy 5, we see there is a potential for the use of a *throw-away* simulation model for all the other four strategies. We contend that the use of simulation should be argued and justified on a case-to-case basis, whether we have a problem that can be explored with the use of simulation given the cost of simulation (time taken to develop, verify and analyze the simulation results), accuracy and reliability of results (subject to availability and quality of data) and the time frame of the VSM process.

In this study, we employed *"Strategy 4"*, to use simulation to support the reasoning and choice of alternatives for improvement to derive the future state map for the products with some success (based on the feedback of practitioners). Unanimously, practitioners indicated that the use of simulation led to more insightful discussions. However, based on their feedback, practitioners will not solely rely on the results of the simulation model. Of the various purposes for simulation identified in the literature (see Chapter 2), in the context of VSM and the case company, we found that due to the lack of dedicated measurement programs to calibrate simulation models, SPSM can be used as a means to reflect on our choices and have a testbed to do a sanity check of our proposed interventions under the given assumptions and preconditions.

Furthermore, based on the feedback from the practitioners, we are less likely to see a transfer of technology when it comes to software process simulation. There is keen interest in the simulation models and other avenues where they will like to use simulation models however the interest at least in the case company stopped at the use

of the models. As shown in these case-studies, simulation can be useful, however it is likely to remain a tool for the use of researchers and consultants who can identify appropriate problems where the cost/time for the use of simulation can be reasoned about on a case to case basis.

Simulation supported VSM certainly gained practitioners' confidence in the improvements. They considered them more realistic and thus considered them more likely to be implements. Satisfactory progress towards implementing the improvements was observed in the follow-up workshops six months after the VSM activity.

Positive feedback on the VSM process that was followed in these two cases, the participants' satisfaction with the outcomes of the study, impact on the case organization in the form of new reports in the dashboard and the adoption and training of other sites to utilize VSM, and further application of VSM on six other products indicates a positive outcome/impact of the studies reported in this chapter.

References

[1] F. A. Abdulmalek and J. Rajgopal, "Analyzing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study," *International Journal of Production Economics*, vol. 107, pp. 223–236, 2007.

[2] K. Agyapong-Kodua, J. O. Ajaefobi, and R. H. Weston, "Modelling dynamic value streams in support of process design and evaluation," *International Journal of Computer Integrated Manufacturing*, vol. 22, pp. 411–427, 2009.

[3] A. Al-Emran, P. Kapur, D. Pfahl, and G. Ruhe, "Studying the impact of uncertainty in operational release planning - an integrated method and its initial evaluation," *Information and Software Technology*, vol. 52, no. 4, pp. 446–461, 2010.

[4] ——, "Supporting information for the simulation assisted vsm," 2014. [Online]. Available: http://www.bth.se/com/nal.nsf/pages/vsm_spsm

[5] S. K. H. Al-Khafaji and H. M. R. Al-Rufaifi, "A case study of production improvement by using lean with simulation modeling," in *Proceedings of the 3rd International Conference on Industrial Engineering and Operations Management, 3-6 July 2012*, 2012, pp. 271–279.

[6] O. Balci, "A life cycle for modeling and simulation," *Simulation*, vol. 88, no. 7, pp. 870–883, 2012.

[7] M. Bevilacqua, F. E. Ciarapica, and G. Giacchetta, "Value stream mapping in project management: A case study," *Project Management Journal*, vol. 39, pp. 110–124, 2008.

[8] A. Bhagat, S. Wang, M. T. Khasawneh, and K. Srihari, "Enhancing hospital health information management using industrial engineering tools," in *Proceedings of the Society for Health Systems Conference and Expo 2009, April 1, 2009 - April 4, 2009*, 2009.

[9] N. L. Coppini, L. C. Bekesas, E. A. Baptista, M. Vieira Junior, and W. C. Lucato, "Value stream mapping simulation using ProModel software," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 6-9 Dec.*, 2011, pp. 575–579.

[10] M. Dotoli, M. P. Fanti, G. Iacobellis, and G. Rotunno, "A lean manufacturing strategy using Value Stream Mapping, the Unified Modeling Language, and discrete event simulation," in *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), 20-24 Aug.* IEEE, 2012, pp. 668–673.

[11] A. Esfandyari, M. R. Osman, N. Ismail, and F. Tahriri, "Application of value stream mapping using simulation to decrease production lead time: A malaysian manufacturing case," *International Journal of Industrial and Systems Engineering*, vol. 8, pp. 230–250, 2011.

[12] B. B. N. de França and G. H. Travassos, "Reporting guidelines for simulation-based studies in software engineering," in *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering, EASE 2012, Ciudad Real, Spain, May 14-15, 2012.*, 2012, pp. 156–160.

[13] B. B. N. d. França and G. H. Travassos, "Are we prepared for simulation based studies in software engineering yet?" *CLEI Electronic Journal*, vol. 16, no. 1, pp. 8–8, 2013.

[14] A. Fuggetta, "Software process: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*. ACM, 2000, pp. 25–34.

[15] S. M. Gahagan, "Adding value to value stream mapping: A simulation model template for VSM," in *Proceedings of the IIE Annual Conference and Expo 2007 - Industrial Engineering's Critical Role in a Flat World, May 19-23*. Institute of Industrial Engineers, 2007, pp. 712–717.

[16] J. Ghosh, "Concurrent, overlapping development and the dynamic system analysis of a software project," *IEEE Transactions on Engineering Management*, vol. 57, no. 2, pp. 270–287, 2010.

[17] M. L. Gillenson, M. J. Racer, S. M. Richardson, and X. Zhang, "Engaging testers early and throughout the software development process: Six models and a simulation study," *Journal of Information Technology Management*, vol. 22, no. 1, pp. 8–27, 2011.

[18] B. Gopakumar, S. Sundaram, S. Wang, S. Koli, and K. Srihari, "A simulation based approach for dock allocation in a food distribution center," in *Proceedings of the 2008 Winter Simulation Conference, Vols 1-5*, 2008, pp. 2750–2755.

[19] A. Gurumurthy and R. Kodali, "Design of lean manufacturing systems using value stream mapping with simulation: A case study," *Journal of Manufacturing Technology Management*, vol. 22, pp. 444–473, 2011.

[20] T. Hall, A. Rainer, and N. Baddoo, "Implementing software process improvement: an empirical study," *Software Process: Improvement and Practice*, vol. 7, no. 1, pp. 3–15, 2002.

[21] W. A. Hamad, J. Crowe, and A. Arisha, "Towards leaner healthcare facility: Application of simulation modelling and value stream mapping," in *Proceedings of the 1st International Workshop on Innovative Simulation for Health Care, September 19-21*, 2012, pp. 149–155.

[22] L. Harjumaa, J. Markkula, and M. Oivo, "How does a measurement programme evolve in software organizations?" in *Proceedings of the 9th International Conference on Product-Focused Software Process Improvement, PROFES 2008 Monte Porzio Catone, Italy*, 2008, pp. 230–243. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69566-0_20

[23] D. Houston, "An experience in facilitating process improvement with an integration problem reporting process simulation," *Software Process: Improvement and Practice*, vol. 11, no. 4, pp. 361–371, 2006.

[24] N. Hsueh, W. Shen, Z. Yang, and D. Yang, "Applying UML and software simulation for process definition, verification, and validation," *Information and Software Technology*, vol. 50, no. 9, pp. 897–911, 2008.

[25] O. Ince, "A simulation study on an office process by applying lean tools," in *Proceedings of the 38th International Conference on Computers and Industrial Engineering*, vol. 1, 2008, pp. 791–794.

[26] J. B. Jensen, S. L. Ahire, and M. K. Malhotra, "Trane/Ingersoll Rand Combines Lean and Operations Research Tools to Redesign Feeder Manufacturing Operations," *Interfaces*, vol. 43, pp. 325–340, 2013.

[27] A. Kasoju, K. Petersen, and M. V. Mäntylä, "Analyzing an automotive testing process with evidence-based software engineering," *Information and Software Technology*, vol. 55, no. 7, pp. 1237 – 1259, 2013.

[28] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling: Why? what? how?" *Journal of Systems and Software*, vol. 46, pp. 91–105, 1999.

[29] M. Khurum, K. Petersen, and T. Gorschek, "Extending value stream mapping through waste definition beyond customer perspective," *Journal of Software: Evolution and Process*, vol. 26, no. 12, pp. 1074–1105, 2014.

[30] Y. H. Lian and H. van Landeghem, "Analysing the effects of lean manufacturing using a value stream mapping-based simulation generator," *International Journal of Production Research*, vol. 45, pp. 3037–3058, 2007.

[31] J.-C. Lu, T. Yang, and C.-Y. Wang, "A lean pull system design analysed by value stream mapping and multiple criteria decision-making method under demand uncertainty," *International Journal of Computer Integrated Manufacturing*, vol. 24, pp. 211–228, 2011.

[32] A. Mahfouz and A. Arisha, "Lean distribution assessment using an integrated framework of value stream mapping and simulation," in *Proceedings of the 43rd Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013, December 8-11*. IEEE Computer Society, 2013, pp. 3440–3449.

[33] T. McDonald, E. M. Van Aken, and A. F. Rentes, "Utilising simulation to enhance value stream mapping: A manufacturing case application," *International Journal of Logistics: Research & Applications*, vol. 5, pp. 213–232, 2002.

[34] H. L. McManus, *Product Development Value Stream Mapping (PDVSM) manual*, 1st ed., Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 2005.

[35] J. M. Morgan and J. K. Liker, "The toyota product development system," *New York*, 2006.

[36] S. Mujtaba, R. Feldt, and K. Petersen, "Waste and lead time reduction in a software product customization process with value stream maps," in *Proceedings of the 21st Australian Software Engineering Conference (ASWEC)*. IEEE, 2010, pp. 139–148.

[37] E. Paikari, G. Ruhe, and P. H. Southekel, "Simulation-based decision support for bringing a project back on track: The case of RUP-based software construction," in *Proceedings of the International Conference on Software and System Process, ICSSP , June*, 2012, pp. 13–22.

[38] M. Paju, J. Heilala, M. Hentula, A. Heikkila, B. Johansson, S. Leong, and K. Lyons, "Framework and indicators for a sustainable manufacturing mapping methodology," *Proceedings of the 2010 Winter Simulation Conference*, pp. 3411–3422, 2010.

[39] C. Pepe, D. Whitney, E. Henriques, R. Farndon, and M. Moss, "Development of a framework for improving engineering processes," in *Proceedings of the 18th International Conference on Engineering Design*, 2011, vol. 1, pp. 417–428.

[40] K. Petersen, "Is lean agile and agile lean?" *Modern Software Engineering Concepts and Practices: Advanced Approaches*, p. 19, 2010.

[41] K. Petersen and C. Gencel, "Worldviews, research methods, and their relationship to validity in empirical software engineering research," in *Proceedings of the Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, Oct 2013, pp. 81–89.

[42] K. Petersen, M. Khurum, and L. Angelis, "Reasons for bottlenecks in very large-scale system of systems development," *Information and Software Technology*, vol. 56, no. 10, pp. 1403–1420, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2014.05.004

[43] K. Petersen and C. Wohlin, "Measuring the flow in lean software development," *Software: Practice and Experience*, vol. 41, no. 9, pp. 975–996, 2011. [Online]. Available: http://dx.doi.org/10.1002/spe.975

[44] K. Petersen, C. Wohlin, and D. Baca, "The waterfall model in large-scale development," in *Proceedings of the 10th International Conference on Product-Focused Software Process Improvement, PROFES Oulu, Finland, June 15-17, 2009.*, 2009, pp. 386–400.

[45] M. Petre, ""no shit" or "oh, shit!": responses to observations on the use of UML in professional practice," *Software & Systems Modeling*, pp. 1–11, 2014.

[46] M. Poppendieck and T. Poppendieck, *Lean software development: an agile toolkit*. Addison-Wesley Professional, 2003.

[47] S. Ramakrishnan, L. Al-Fandi, and J. Chen, "A simulation based framework to study the impact of lean techniques on green supply chains," in *Proceedings of the 30th Annual National Conference of the American Society for Engineering Management, ASEM , October 14-17*, 2009, pp. 539–546.

[48] C. Robson, *Real word research*. Oxford: Blackwell, 2002.

[49] A. Rohana, O. Nooririnah, H. Isa, S. R. Kamat, and M. P. Mehad, "Lean waste analysis and improvement using dynamic value stream mapping," *Global Engineers and Technologists Review*, vol. 3, pp. 1–8, 2013.

[50] M. Rother and J. Shook, *Learning to see: Value stream mapping to add value and eliminate MUDA*. Brookline, Mass.: Lean Enterprise Institute, 1999.

[51] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[52] M. A. Shararah, "A value stream map in motion," *Industrial Engineer: IE*, vol. 45, pp. 46–50, 2013.

[53] M. A. Shararah, K. S. El-Kilany, and A. E. El-Sayed, "Value stream map simulator using ExtendSim," in *Proceedings of the World Congress on Engineering (WCE), 6-8 July 2011*, 2011, pp. 755–758.

[54] D. Sjøberg, A. Johnsen, and J. Solberg, "Quantifying the effect of using kanban versus scrum: A case study," *IEEE Software*, vol. 29, no. 5, pp. 47–53, Sept 2012.

[55] P. Solding and P. Gullander, "Concepts for simulation based value stream mapping," in *Proceedings of the Winter Simulation Conference*, 2009, pp. 2172–2178.

[56] C. K. Swank, "The lean service machine," *Harvard Business Review*, vol. 81, no. 10, pp. 123–130, 2003.

[57] S. Vinodh, M. Somanaathan, and K. R. Arvind, "Development of value stream map for achieving leanness in a manufacturing organization," *Journal of Engineering, Design and Technology*, vol. 11, pp. 129–141, 2013.

[58] X. Wang, K. Conboy, and O. Cawley, ""leagile" software development: An experience report analysis of the application of lean approaches in agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1287 – 1299, 2012.

[59] J. Williford and A. Chang, "Modeling the fedex {IT} division: a system dynamics approach to strategic {IT} planning," *Journal of Systems and Software*, vol. 46, no. 23, pp. 203 – 211, 1999.

[60] W. Xia and J. Sun, "Simulation guided value stream mapping and lean improvement: A case study of a tubular machining facility," *Journal of Industrial Engineering and Management*, vol. 6, pp. 456–476, 2013.

[61] Q. Yang, X. Chen, and T. Lu, "Case study: Simulation analysis for complex r&d project based on iteration process model," in *Proceedings of the 2011 International Conference on Instrumentation, Measurement, Circuits and Systems*, 2011, pp. 57–60.

[62] Q. Yang, X. Chen, and Z.-l. Yan, "The Lean Improvement Analysis based on Simulation for the Process of Civil Aviation Service," *Industrial Engineering and Management*, vol. 15, pp. 51–56, 2010.

[63] X. Yikun and P. Qingjin, "Integration of value stream mapping and agent-based modeling for OR improvement," *Business Process Management Journal*, vol. 18, pp. 585–599, 2012.

Chapter 6

FLOW assisted value stream mapping in a large-scale software product development

Abstract

Value Stream Mapping (VSM) has been successfully applied in the context of software process improvement. However, its current adaptations from Lean manufacturing focus mostly on the flow of artifacts and have taken no account of the important information flows in software development. A solution specifically targeted towards information flow elicitation and modeling is FLOW. This study aims to propose and evaluate the combination of VSM and FLOW to identify and alleviate information and communication related challenges in large-scale software development. Using case study research, FLOW assisted VSM was used for a large product at Ericsson AB, Sweden. Both the process and the outcome of FLOW assisted VSM have been evaluated from the practitioners' perspective. It was found that FLOW helped to systematically identify and elaborate wastes, challenges and improvements related to information flow. Practitioners were positive about the use of VSM as a method of choice for process improvement and found the overview of the entire process using the FLOW notation very useful. The combination of FLOW and VSM presented in this study was successful in uncovering issues and systematically characterizing their solutions, indicating their practical usefulness for waste removal with a focus on information flow related issues.

6.1 Introduction

Value stream mapping (VSM) is a lean practice that maps the current product development process (CSM), identifies value adding and non-value adding activities and steps, and helps to create a shared action plan for an improved future state for the process i.e. a future state map (FSM) [4] [11] [5] [9]. VSM looks at both material and information flow [11]. In software product development, an equivalent analysis of material flow will look at the flow of *"work items"* e.g. a requirement, use case or a user story etc. through the process. Contrary to simply the scheduling information that is captured in terms of "information flow" for production processes, for software development it will be pertinent to capture documented/verbal, formal and informal communication. Furthermore, the information needs, knowledge and competence need to be identified and analyzed to facilitate the value adding activities in the software development process.

Traditionally, VSM has had a high focus on mainly tracing requirements or other work items through the process, identifying waiting and productive times [5] [4] [6]. No particular challenge occurs here if the communication structure is relatively simple. However, as the situation becomes more complex (as was the case in the studied organization) it becomes equally important to focus on the information flow and explicate it. The existing representation of VSM [5] and [11] does not allow capturing the flow of communication. As a consequence, we cannot identify value-adding and non-value adding activities required to streamline the process of value creation.

Information flow modeling (FLOW) has been proposed as a systematic method to analyze and improve communication in software development processes by considering all types of experience and information flows in the organization [15].

We propose to combine value stream analysis with FLOW. As both share a focus on capturing and improving information flow, their combination can yield potentially useful results. FLOW applies a systematic approach and a structured yet simple graphical notation to identify and capture the flow of information between people, documents, and activities. It covers and combines formal, informal, documented and verbal information flows in a complex communication structure; hence, it can compensate the inability of existing VSM notation to capture this aspect of information flow in software product development. Previously, FLOW has been successfully used in software process improvement in a variety of smaller-scale software development units to capture and improve the communication structures in industrial settings [15].

To evaluate the usefulness and scalability of using FLOW to assist VSM, we conducted case study research in a large-scale product development at Ericsson AB. At the time of this study, the product under investigation was a solution comprising 12 subsystems, the number has since then grown even further. The development unit has over 2500 employees in 10 different centers of excellence, which are located in Germany,

Sweden, India, China, Canada, and the US. We conducted six value stream workshops, with two researchers (first two authors) acting as facilitators, and the third author (who is the inventor of FLOW) as an external observer.

The remainder of the chapter is structured as follows: Section 6.2 presents related work. Section 6.3 describes the research method. Section 6.4 presents how FLOW can be used to assist in a typical value stream mapping activity. Section 6.5 presents the results and Section 6.6 reflects on the obtained results. The results of a follow-up six months after the conclusion of the study are presented in Section 6.7. Section 6.8 concludes the chapter.

6.2 Related work

The related work for this study has two main themes: first is the use of VSM and second is the use of FLOW for mapping and improving software development process.

6.2.1 Value stream mapping

Rother and Shook [11] presented the guidelines to conduct VSM for manufacturing processes where both material and information flow are mapped. Realizing the differences in product development and manufacturing McManus [5] adapted VSM for this new context. McManus [5] developed a manual for applying VSM for product development which is the foundation of this work along with its extension by Khurum et al. [4].

Two applications of VSM were reported in the context of software product development [6], [4]. Mujtaba et al. [6] have reported a VSM for product customization process where the goal was to reduce the lead time. Khurum et al. [4] identified value aspects that are not traditionally considered, but have to be evaluated on a case to case basis, based on what each organization conducting the VSM consider important from their customers' perspective. In both cases, VSM was done on mature products that have been on the market for a number of years. In this study, we apply VSM in the context of a new large-scale software product development distributed across sites in multiple countries.

Poppendieck and Poppendieck [9] also provide a brief three step process to do a VSM. However, the overwhelming focus of analysis in these guidelines is on developing a time-line of progress through a process and making a distinction between waiting and productive time (c.f. [9] [6] [4]). The aim of lean is to ensure that a process should make only "what" is required and "when" it is required by the next process [11]. The

time-line analysis focuses primarily on “when” in the aim above and seems to overlook the “what” aspect i.e. what information needs exist that should be fulfilled.

In this study, we complement the time line based analysis of work items, which is typical in VSM, with the use of FLOW methodology to identify and analyze information flows in the case of large-scale software product development.

6.2.2 FLOW

Through this distinction and visualization of the information flows in the development process using a modeling notation, FLOW allows the identification of sub-optimal paths for communication [16]. The simple notation used by FLOW is presented in Figure 6.1. The original FLOW notation as presented by Schneider et al. [14] and used in previous industry applications, uses slightly different symbols for documents and persons. A document is usually represented by the document symbol available in PowerPoint; a person is represented by a Smiley. Both symbols are readily available and do not require any specific editor. The stick figures and document symbols used in this study are more familiar to the industry participants. This advantage exceeded the conformance with previous applications of FLOW.

Many established software process models focus on documents only [10]. Verbal or informal communication, and the use of experience in sophisticated activities are, however, often neglected. In industrial reality, however, many requirements, decisions, and rationales are communicated in an informal way, via phone, meetings, or personal email. The FLOW method offers a graphical notation (Figure 6.1) and a modeling approach that covers and combines both documented (solid) and non-documented, verbal, or informal (fluid) flow of information. Traditional plan-driven software development

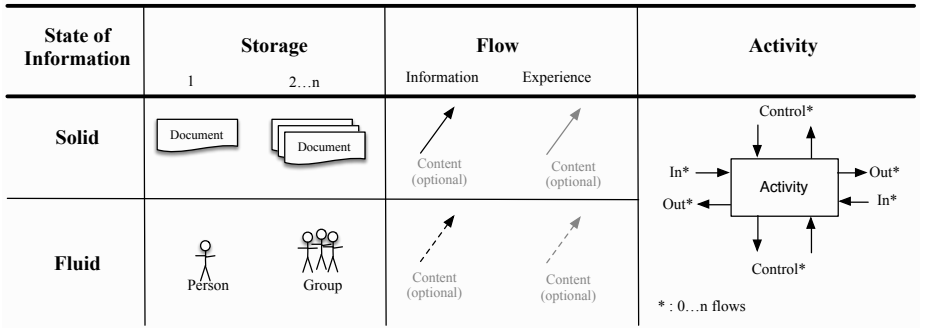


Figure 6.1: Symbols used in the graphical FLOW notation.

tends towards solid information propagation, whereas agile or flexible approaches rely on direct communication [3], of fluid information.

Both solid and fluid information flows have complementary advantages and weaknesses. Solid information is typically stored in documents, files, recordings, or other kinds of permanent media. Creating and retrieving information takes longer and requires more effort, but does not rely on the creator, and can be performed at a later point in time and by many people. Fluid information flows faster and often more easily: phone calls, chats, or short messages do not get formally documented, but convey information as well. Fluid information is typically stored in a person's mind. Therefore, a face symbol is used to visualize it.

Many companies need to mix and adjust elements from both ends of the spectrum and try to find an overall optimum. FLOW was created to capture, visualize, and improve situations consisting of a complex network of fluid and solid information flows. Experience is considered an important special case of knowledge that must be present when difficult activities occur and important information is routed through the organization [16]. It is also integrated in the FLOW method and notation (Figure 6.1).

The notation is intentionally kept very simple. It is supposed to be used on a whiteboard or a piece of paper to discuss current and desired situations. An activity is symbolized by a rectangle with several incoming and outgoing flows. The rectangle acts as a black box and hides internal details of the activities and flows. It can be refined by another FLOW diagram [13].

At the beginning of a FLOW analysis, the current communication channels and network of information flows must be identified. The FLOW method [16] [17] offers a technique based on interviews. This technique has been applied to medium-sized groups in several companies. Rather complex networks of solid and fluid information and experience flows were identified, modeled, and discussed with domain experts. In one case, a repository was mined as an indicator of document-based (solid) information flows to complement interview results [20]. The FLOW elicitation technique has not yet been applied to elicit and model complex information flows in very large organizations. Building the model is the first step towards improving the overall information flow. FLOW interviews were optimized for individuals. An important research question was, therefore, whether the interview technique could be transferred directly to a larger group of people (workshop), and how it needed to be adapted.

Several researchers have investigated dependencies and dynamics of software projects including the impact of information flowing through projects: Pikkareinen et al. [8], for example, investigate communication in agile projects and their impact on building trust. The focus of their work is on the impact of agile communication patterns on project success.

Winkler [18] uses the term information flow, too. He focuses on dependencies between artifacts. An artifact is a part of a document. His main interest is traceability of requirements. Winkler considers only document-based requirements and information flows. He uses different ad-hoc illustrations to discuss flow of requirements. Berenbach and Borotto [2] present requirements project metrics. All metrics are based on solid information only. However, information flow within their Formal Requirements Development Process could be modeled in FLOW. Some of their metrics could be extended to refer to fluid information, too.

In principle, information flow can be seen as an aspect of a process. Process models such as BPMN or BPEL or workflow models such as Little-JIL [19] would, therefore, be an alternative to FLOW models. However, the scope of FLOW models is smaller, and the focus on fluid and solid information is more specific than in those models that are more general. In addition, the symbols of a FLOW model were carefully selected to visualize human and document-based stores of information. Process models tend to emphasize sequence and coordination of control flow more than the flow of information. The latter was considered more important for this work.

Just as we argued that FLOW supports information flow analysis in the VSM, FLOW can benefit from VSM's systematic way of reflecting on wastes and identifying improvements. Besides the systematic approach, the time-line based analysis can help quantify the implications of challenges in information flow in software development. Hence, both VSM and FLOW have synergies and may be even more successful in combination.

6.3 Research methodology

This study aimed to use and assess FLOW assisted VSM for the improvement of large-scale software product development process. In particular, to assess the ability to capture complex information flows and identify improvement opportunities in the development process.

RQ: *How useful is the combination of FLOW and value stream mapping in large-scale software development?* The following sub-research questions were posed to answer it through a case study at Ericsson AB, Sweden, where VSM supported with FLOW was applied and evaluated:

- *RQ1.1: Can the interview based technique for FLOW be transferred directly to a larger group of people (in a workshop setting), if not, how can it be adapted?*
- *RQ1.2: Which wastes and improvements could be identified with the combination of FLOW and VSM?*

- *RQ1.3: How do practitioners perceive the process and outcomes for the VSM supported by FLOW?*

6.3.1 Context

The case company is Ericsson AB, Sweden, which is an ISO 9001:2000 certified telecommunication vendor. The product that was studied is still under development and has not had a release to the customer yet. Overall, there are approximately 12 systems that constitute this product. The development unit has over 2500 employees in 10 different centers of excellence, which are located in Germany, Sweden, India, China, Canada, and the US. Teams use agile principles and practices in development, e.g. writing user stories, working in fixed length sprints (time boxing), and cross functional teams, etc. The overall process is conceptualized in Figure 6.2.

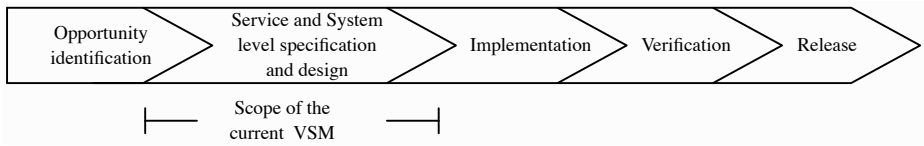


Figure 6.2: Overview of the product development process in the case company.

The process starts with the identification of an opportunity for the organization to exploit e.g. a new potential market, increased customer retention if the product is compliant to certain standards etc. It is formulated as a Business Opportunity (BOP). Based on the business case, a decision to persist further with the development of a BOP is taken.

High-level analysis of the accepted BOPs with the consideration for the product anatomy, technical requirements and the business context *BOPs* is performed, which results in more detailed specifications in the form of one of more business use cases (BUCs).

BUCs are the work-item used to plan, control and manage development process in the company. A detailed analysis of BUCs is performed and the systems that are required to implement the BUC are identified. A further detailed analysis by the system teams involved in the implementation of a BUC produces more detailed system level artifacts called business sub-use case (BSUCs). At this level, the system teams take on the responsibility of the implementation of the BSUCs for their systems.

The complexity and large scale of the product development entailed that it was impossible to cover the entire process in sufficient detail to analyze it for the chal-

lenges and improvements (with 20 systems and also the number of geographical sites involved).

Thus, it was decided to limit the current VSM activity to the *"design and specification phase at both BUC and BSUC study level"* of the overall product development process. Furthermore, this phase is of critical importance as the allocation of work and coordination to reach the products goals through individual systems (which are developed in geographically distributed sites) depends on this phase and currently the organization is facing a lot of challenges in this phase. Lastly, as it is a new product, none of the BUCs have reached the verification phase and beyond (see Figure 6.2).

The teams primarily responsible for the specification phase at BUC level are located in the chosen site. One of the system development teams, responsible for BSUC level analysis, is also collocated at this site. Thus, we were able to cover the entire specification phase with coverage of both the service, i.e. BUC, and system, i.e. BSUC level in the chosen site.

Thus, due to the scale of the product and the reasons listed above it was not possible to start with an end-to-end VSM of the entire development process. However, after the completion of this study, VSM was conducted for several of the individual sites to get an end-to-end perspective.

6.3.2 Data Collection

The two main sources of data were: the five workshops that were conducted as part of VSM and the corporate data on work items. This data included release plans, number of BOPs, BUCs and BSUCs their status, time spent in each phase, priority, number of impacted systems, initial size estimates, categorization as function or quality requirements. We relied on extensive note taking, pictures of the white board and collection of yellow stickers for collecting data during the workshops. Notes from the researchers were corroborated and compiled into reports before the subsequent meeting where a summary was presented.

The first two authors of the study were acting as VSM facilitators who were responsible for data collection, moderating workshop sessions, analyzing the collected information and presenting the results. The third author served as an expert on FLOW and observed and discussed the process that was followed during the VSM.

One practitioner served as the "VSM Manager" who had the upper management support and could sanction the resources and time necessary to successfully complete the VSM activity. Five other practitioners were involved who were present during the workshops. These practitioners were chosen due to their knowledge and experience of the process and also their stakes in it (their roles in the development process and aptness for the workshops is discussed further in Section 6.5.1).

Furthermore, data about progress tracking for BUCs and BSUCs was extracted by individual system owners and was given to the researchers in spreadsheets. A relational database was created to reconstruct the links between the BUCs and their break down at system level BSUCs. Furthermore, examples of BUC and BSUC output were analyzed to triangulate the practitioners' opinion and perception of the quality of the artifacts in terms of their content and level of detail.

6.3.3 Validity threats

This section reports the validity threats to the findings of this study and what measures were taken to alleviate them. The limitations of the study are also discussed here.

Reliability related threats influence the repeatability of a study e.g. dependence of the research results on the researchers who conducted it would diminish the reliability of a study [12]. To avoid inaccuracy in captured information (information and views captured during the workshops), two researchers took notes while the third researcher was moderating the workshops. These notes and their interpretation were triangulated among the researchers. Also, the results of the previous workshop were presented to the practitioners at the start of each subsequent workshop to identify any misinterpretations by the researchers.

Internal validity is dependent on researchers' awareness or ability to control the extent of a factor's effect when investigating a causal relation [12]. Internal validity of the results was increased by the involvement of multiple practitioners having various roles and responsibilities in the development process. Multiple data sources were used in this study, the expert opinion on challenges where possible was triangulated with actual process metrics e.g. the perception of under-estimation of required effort was validated by comparing the estimated time and the actual time spent on analysis. Similarly, number of closed BUCs and BSUCs were used to validate the frequent reprioritization and quality of initial analysis.

Ericsson promotes continuous improvement in software development at all levels. We are aware that some participants in VSM related workshops were participating in other ongoing improvement initiatives. Thus, there is a risk that the challenges and improvements identified in this study cannot be solely attributed to the intervention in this study. Their involvement in multiple initiatives would have indeed influenced some findings of this study as well. But due to the different focus of the initiatives, while there were some overlaps between the outcomes of the study, a majority of the participants agreed that the use of FLOW assisted VSM lead to new insights into the process that they did not have before.

Construct validity reflects whether the measures used by researchers indeed represent the intended purpose of investigation [12]. The opinions of practitioners on the

process and outcomes of the VSM were collected using a questionnaire. To reduce the risk that questions might be misunderstood, the researchers were present during this session to answer any clarification questions. To reduce the influence that our presence may introduce, and to encourage honest opinions, practitioners were asked not to put any identification information on the filled questionnaires. Furthermore, using a semi-structured format, complementary themes were explored in a group discussion after the practitioners had filled the questionnaire.

External validity is concerned with the generalization of the findings of the study outside the studied case [12].

Both methodologies, FLOW and VSM, have been individually validated in industrial studies. While their proposed combination has only been studied here on one product, however, it has been shown that the use of FLOW successfully addresses the limitations of VSM. The context of the study, the process of combining FLOW with VSM and under what conditions this combination becomes useful has been defined in sufficient detail for other practitioners to judge the relevance for their unique context.

6.4 FLOW assisted VSM

An important aim for lean is to ensure that one process makes only what is required and when it is required by the next process [11]. To ensure this VSM considers both material and information flow. In software development context, material flow is analogous to tracing work packages (like requirements, use-cases etc.) through the process. However, the information flow is much more than just the timing/scheduling information in production processes. It includes documented and undocumented channels of information flow through which formal and informal exchange of information takes place [16].

In this regard, the conventional VSM notation [11] is insufficient to capture the details of the knowledge intensive software development processes. This limitation has not been addressed in the adaptation of VSM for product development VSM either where the additional information captured is just regarding the iterations [5].

As discussed in Section 6.2 FLOW is a methodology that systematically captures and visualizes channels for information flow [16]. Its simple notation, ability to identify and visualize both documented and undocumented channels of information, make it a practical and relevant complement for use in conjunction with VSM. The simplicity of the notation makes it more scalable compared to more complex modeling notations. A FLOW model mainly depicts the network of information flow (people, documents, etc.) and the types of channels used.

VSM has five distinct steps, Figure 6.3 lists the steps and the main activities in each step (for details please see [4]). FLOW can be used in step S2, to elicit the CSM. However, we consider that it may not be necessary in every case to use FLOW for establishing a CSM. For example, if the challenges are related to long waiting times for items in backlogs, redundant reviews etc. then it may not be necessary to model different types of information, stakeholders and competence. So, depending on the challenges identified in step S3 (see Figure 6.3), we can decide whether the use of FLOW is justified. Thus, if there are challenges related to knowledge management or information flow, as was the case in this case study, then there is a utility of FLOW to identify the information needs, bottlenecks, broken communication paths and thus help in eliminating waste and creating a FSM in step S4.

In particular, the amount of information and its variety grows with the number of persons/sites involved, hence this has to be understood as well. Also, the information flows have to be understood to help explain why certain waiting times (i.e. the time when "work items" that move through the process are inactive) exist in a process.

In the context of VSM, we can benefit from FLOW in the following ways:

- First we can use it for eliciting the information flows in the product development process.
- Secondly, its graphical notation can be used to represent the current state map.
- Lastly, it can be used to identify and reason about changes in the current process to derive and document the FSM with improvements.

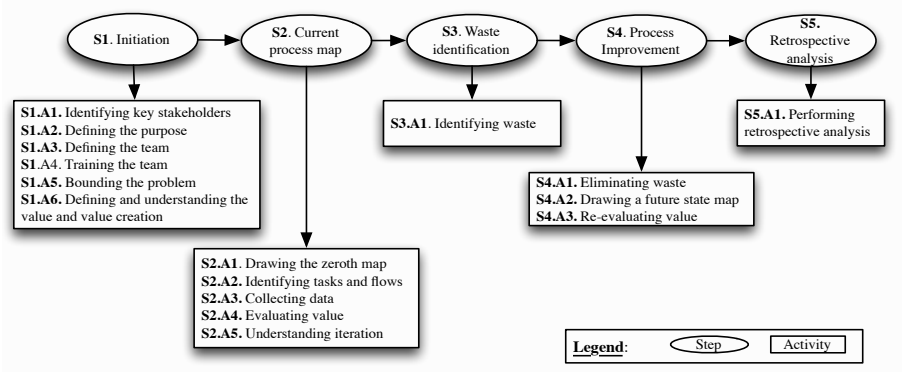


Figure 6.3: Overview of the VSM process based on [4].

Five aspects listed in Table 6.1 are elicited for each activity in the FLOW elicitation technique. They refer to the incoming and outgoing flows of an activity (see Figure 6.1: activity symbol) and ask for the details of each flow.

In this research, we used FLOW to materialize the lean principle of pull" for information-flow in the product development process. The idea is to use the pull-based approach starting from the down-stream phase and identifying what information and experience are required from an upstream phase of the process. In previous applications, the order of FLOW interviews did not consider the position of activities in the development process. Instead, the availability of interviewees was the primary concern.

Within FLOW interviews, the order of flows around one activity is usually:

- What are the activities you are involved in? Each one is detailed afterwards.
- What does this activity produce, and where do they go? The out-going flows respond to the pull of down-stream activities.
- What input is needed in order to produce this output, and where do you get it from? Very similar to data flow diagrams, there must be a source (incoming flow) of every information.
- Who helps or directs the activity? The activity may be controlled and supported by checklists, guidelines, or experience. It is equally important to model where that experience is supposed to come from even if it is a person bringing this experience to the table (fluid experience).
- What tools or support do you need for performing the activity?

The questions are designed to cross-check incoming and outgoing information with the receiver or source, respectively. Inconsistent answers indicate either inconsistent terminology (spec vs. requirements document), incomplete knowledge of the source or

Table 6.1: Elicitation tool based on FLOW used for each activity

Aspect	Questions
Tasks	What are the main tasks in this activity? e.g. write user stories
Roles	Who should be involved in this activity?
Output	What is the output of the activity? e.g. test cases
Intput	What is the input to the activity? e.g. user stories
Content	Details of the input required?
Who/What	Who has the information needed for this activity? What is the source of information?
Competence	What competence/knowledge sources are required to facilitate this activity? e.g. architectural expertise, knowledge repository

target, or misunderstandings among co-workers, e.g. as one of the practitioners said: *(I always thought our BUC would be used by system teams as high-level input to do a detailed analysis).*

While eliciting information for each aspect in Table 6.1 a distinction should be made between the current practice and the ideal practice e.g. when covering the content aspect, two set of questions should be posed:

1. What information do you need? The aim here is to identify the desired state of the process.
2. What information do you currently get? This question aims to explore the current state.

Once this has been done for all the aspects of each of the activities in the CSM, one may compare the current state with the desired state to identify what is unnecessarily produced and what is missing. Similarly, having identified the personnel involved in each activity we can identify situations where there is an information overload on some key personnel. Other critical questions could include reflection on whether we want certain channels to be fluid or solid. The strengths and weaknesses of each type must be balanced, given that we are now looking at a globally distributed context with geographical and temporal distance can we avoid documenting the decision about the priority of BUCs in a local spread sheet.

6.5 Results

In this section, we report the results of applying the five step VSM process assisted by FLOW (as presented in Section 6.4).

6.5.1 Initiation

Since the product is relatively new (no BUCs have gone beyond the implementation phase) and very large, it was decided to reduce the scope of this VSM activity to the "specification" phase of the process . Another motivation was that all the necessary stakeholders for this phase were located in Sweden. The specification phase is responsible for formulating the work package which takes the form of a *Business Use Case* (BUC) based on the business opportunities (BOPs) that have been identified (see Section 6.3.1 for an overview). The practitioners involved in the specification phase were also involved in a high-level analysis to identify dependencies, set priorities and indicate which individual systems would be impacted and therefore should be involved in further analysis and design of the BUC.

After negotiation with the product management the VSM goal for this product was to *"reduce the lead time for end-to-end development process with a focus on improvements in the specification phase"*.

Besides the scale and other challenges listed above, we wanted to analyze the end to end development process to wasteful information that is not really required in the process downstream. So, we started with the "release" phases (see Figure 6.2 for an overview of the process) and wanted to "pull" the information needs from there onto the phases upstream. However, this proved too ambitious due to the size and complexity of systems involved in the product.

Therefore, we not only had to revert to the original scope of the VSM activity, but we also started with the first activity in the process that is the specification of a BOP and identified what information should it contain and what competence is required to define BUCs from it and we continued so on and so forth for rest of the activities in the specification phase.

We started the other way around as to refine the things, it was easier to start from the beginning. At the start, the information need is more simple and on a much higher abstraction level and that made asking what needs to be added, at least for this complex case work better.

Table 6.2 lists the roles and experience of the participants in the VSM workshops. The participants were chosen with the scope of the process and diversity of roles in mind for the following reasons: 1) No practitioner alone knows all the pertinent details for the end-to-end process 2) their involvement helps to develop a consensus on the current way of working, 3) identifying challenges as even if there are individuals well versed with the process they may not be aware of the challenges that the people working in the phase have to face, 4) prioritizing of challenges, otherwise we may sub-optimize by concentrating on a phase and not considering the end-to-end target 5) help to get the commitment on improvement actions and plan.

6.5.2 Current state map

In first workshop with all the participants, an introduction to VSM, its process, experience of using it (especially in the case company), the goal for the current VSM activity and the rationale behind it were briefly presented. VSM Manager specifically related the company's goals and to the goal of the VSM and encouraged participation in the workshops as an opportunity to effect the future way of working.

After the introduction, we start with the "zeroth map" which is a very high-level view of the product development process in the company (as discussed previous in Section 6.3.1). Then the participants were asked to draw the current process map on the board by following the work package i.e. a Business Use Case (BUC) through

Table 6.2: Participants of the VSM workshop

Role and responsibilities	Experience with the product	Experience with the company
Line manager, part of core team, system management	1.5	14
System manager, working on service-level analysis and system design	1.5	19
Program manager, budgets, priority settings, staffing requests	0	21
Architect, System design, requirements	3	20
Project Manager	0.6	14

the process (in this case from "opportunity identification" till the development starts on the respective BUCs see Figure 6.2). With group participation, one typical BUC is followed through the process and the activities and tasks in the current process are mapped. The outcome of CSM from the first workshop is presented in Figure 6.4.

The process starts with the high-level analysis and breakdown of BOPs into BUCs by the core team. Then a team lead by a BUC author consisting of test manager, project manager and representatives from impacted system teams perform detailed analysis (called *BUC study*) on individual BUCs. A core team contact person is used if there is a need for further elaboration on BOP or to assess if the detailed BUC specification is in line with the market need. BUCs cover service level concerns that is inter-system interactions etc. These BUCs are then assigned to dependent system teams that perform *BSUC studies* to address system level concerns and ensure that various system development teams are able to develop and test in a coordinated manner. This is a typical process view that will be a starting point for VSM. However, in the next section when we look at the challenges (Table 6.3) this view is insufficient to address them.

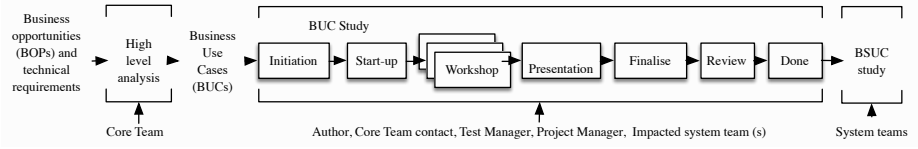


Figure 6.4: Current state map

6.5.3 Challenges

Both qualitative and quantitative information was used to identify challenges in the current process.

Expert opinion

Once the CSM was done, the participants were asked to reflect on the challenges that impede the product from reaching the goal of reducing the lead-time. To encourage participation, each practitioner was asked to write down at least three challenges on *"sticky notes"*. One by one, they were asked to read out the challenges they have listed, to give a brief description of the challenge and to place it on the CSM. The next participant would do the same and if there was an overlap with the already identified challenge they will be put the *"sticky notes"* together. This was supplemented by VSM facilitators' notes of challenges that were mentioned by the participants earlier while creating the CSM. An overlap here could have meant that something is perceived as a greater challenge than others. However, we knew from experience that this only highlighted that a certain challenge was better known and talked about in the company.

Therefore, once the consolidated list of challenges was created, it was prioritized in terms of the impediment or hindrance it poses to achieving the stated goal. The 100 dollar method was used to create a prioritized list of the challenges [1]. Each participant had ten dollars to spend on the list of the challenges, paying more for the challenges they believe are major hurdles to achieving the stated goal. This collective prioritization was useful to reduce the impact of any personal biases of participants and to develop a consensus on what is collectively considered important. The challenges identified, and their relative priorities are listed in Table 6.3.

Moreover, a rudimentary root cause analysis was done for the top four challenges to understand the reasons behind the challenges. These are summarized in Table 6.4.

Given that most of the challenges were related to documentation, understanding information and competence needs it was evident that existing notation for describing the current state map (as shown in Figure 6.4) will be insufficient. Besides, the typical timing based analysis [7] [5], it was decided to utilize FLOW [16] that systematically captures information flows in the software development process. The current state map (as shown in Figure 6.4) was used as a list of activities that became input for FLOW where details of each activity were further elicited.

Lessons learned: If we, in a CSM find challenges related to: documentation, lack of common understanding, and unsatisfied information needs, it points to the need to conduct information modeling in the context of the value stream mapping analysis.

Table 6.3: Perceived priority in terms of the hindrance these challenges pose to reach the stated goal.

ID.	Challenges	Votes
1.	Lack of high-level analysis and documentation of business opportunity analysis and breakdown to BUCs	12
2.	Lack of documentation/guidelines/template/scope for system-level analysis	10
3.	Lack of a common understanding of what information and the level of detail should be the outcome of a service-level analysis	10
4.	Incomplete architecture models / specification / functional baseline	10
5.	Utilization of correct competence	4
6.	BUC not updated/ Neither treated as an alive document nor have an alternative document to track changes	2
7.	Lack of end-to-end BUC responsibility	1
8.	Incomplete information model	1
9.	Quality of business opportunity document (Requirements instead of Business Opportunities)	-
10.	Challenges in communication with acquired companies having different maturity levels in the Telecommunication domain.	-

Quantitative analysis

Figure 6.5a shows the estimated time required for completing the analysis and the actual time-units taken to complete it. While Figure 6.5b shows the distribution of lead-times for various BUCs for the analysis.

By analyzing the lead-time for this phase, it was identified that unlike the typical initial estimate about the high-level study which should at most take 6 time-units in reality even the median value of actual time spent was higher than that with the worst lead time of 23 time units (see Figure 6.5). This was consistent with the general perception (expressed in the workshops by practitioners), that the initial estimates are very inaccurate and underestimate the required effort.

This may explain another observation about the current way of working, i.e. why the SPM team would not provide a prioritized list of BOPs and assign more work than the capacity of the analysis team (as expressed by practitioners in the workshops and also reflected in the long lead-times). Thus, the mismatch is because of the SPM team's expectation that the analysis is not an effort intensive task (as seen by the consistently under-estimation of effort required as shown in Figure 6.5) while the reality is very different. Furthermore, the consistent under-estimation raise the question, whether the BUCs are broken down to the appropriate scope?

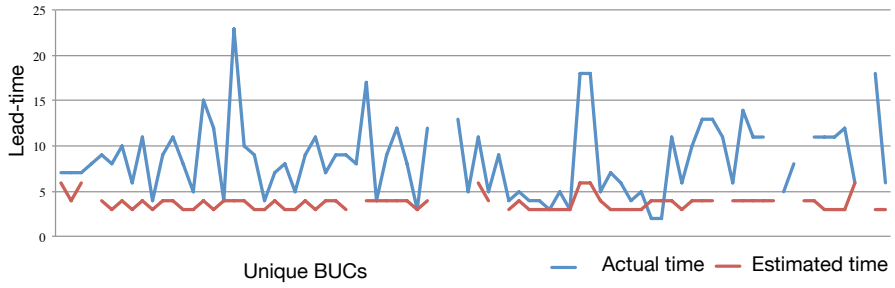
By looking at the release plan and current statuses of various BUCs we were able to identify two wastes:

Table 6.4: Challenges and the reasons behind them.

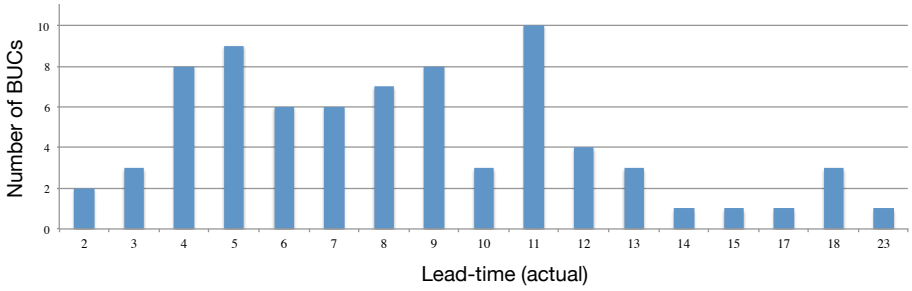
ID.	Challenge	Reasons
1.	Lack of high-level analysis and documentation of business opportunity analysis and breakdown to BUCs	Due to time constraint, there is a lack of documentation with over-reliance on the people in the core team. This sometimes also leads to inconsistent responses from within the core team.
2.	Lack of documentation/guidelines/template/scope for system-level studies	Underestimated the need, had "hoped" that development can start without doing this level of analysis. Also, there is a lack of awareness that this analysis has to be done.
3.	Lack of a common understanding of what information and level of detail should be the outcome of a service-level study	It was decided early on that service-level analysis (BUC study) will be kept on a high-level. However, contrary to this directive the expectations of the system teams responsible for system development are different (they expected much more detailed level of analysis and specification). Between the service-level analysis which is relatively high-level and the system-level analysis which only covers the analysis for intra-system level details, a detailed analysis of inter-system level is missing for BUCs that require more than one system to deliver.
4.	Incomplete architecture models / specification / functional baseline	Underestimated the work, complexity and time it will take. Also, the value of having these for the product was underestimated to facilitate analysis and later on design and development. When reflecting on the causes, practitioners recollected statements like *"we will do it later"* or that the product is *"based on a legacy system so there is no immediate need"* for these documents. Another reason is weak governance i.e. to manage and enforce such standards based on the information needs.

1) Instead of *"build integrity in"* more immediate focus is on delivering functionality: very few i.e. less than 18% of non-functional requirements were assigned to the upcoming release while a majority was assigned to a later release. The approach is roughly visualized in Figure 6.6 where the current focus is on delivering functionality.

2) *"Partially done work"* instead of taking one item through the process: There were slightly fewer ongoing BUCs that were assigned to the subsequent two releases than the number of BUCs where some work has been done. The reason for such a high number was mostly re-prioritization where a lot of work was started in parallel and then the direction changed. In a changing market for a new product, it is highly likely that the priorities will change in the future as well and if this way of working continues that will mean a lot of wasted effort.



(a) Comparison of actual and estimated time required for analysis



(b) Distribution of BUCs based on lead-time

Figure 6.5: Analyzing the estimated and actual time spent on BUC analysis

Similarly, the quality of initial impact analysis (to determine which systems will be involved in the development of a BUC) at Service-Level (i.e. during the BUC study) was assessed in two ways:
First the initial estimated number of systems impacted by a BUC was compared with the actual number of systems eventually involved in the specification phase at the system level (see Figure 6.7).

Secondly, the number of canceled BSUCs was analyzed. This gave the opportunity to reflect whether the impact analysis was working sufficiently well, the cost of re-prioritization where the work done at system-level had to be discarded as well. Figure 6.7 shows that the teams are fairly accurate in predicting the number of impacted systems for BUCs that have a narrower scope. The estimates are relatively inaccurate for BUCs where a larger number of systems were impacted. This analysis can show if there were unnecessary system level (BSUC studies) that were initiated and later closed leading to wasted effort.

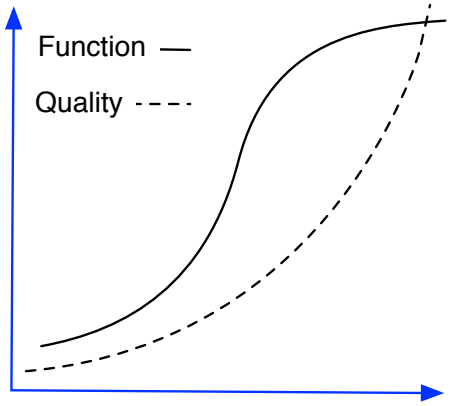


Figure 6.6: The trend in prioritization of non-functional requirements

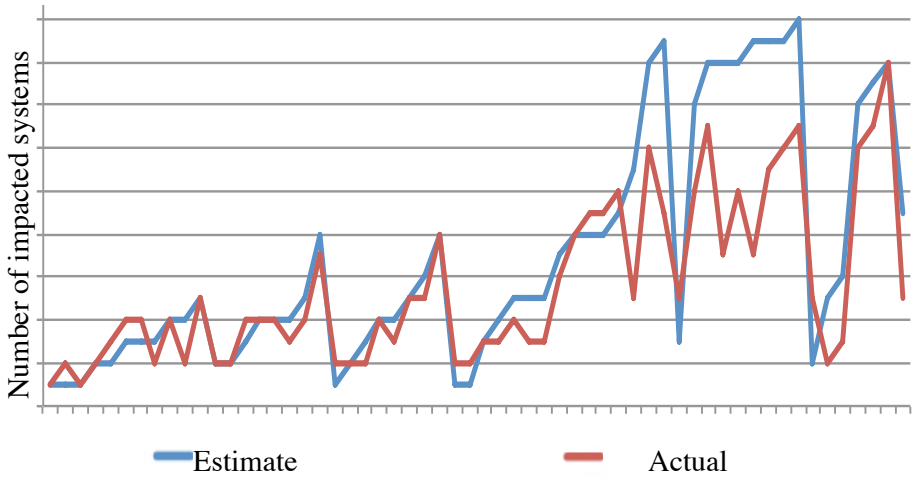


Figure 6.7: Comparison of estimated and actually impacted number of systems by each BUC

6.5.4 Improvements and future state map

Based on the analysis reported in Section 6.5.3 the following considerations were highlighted when creating the future state map and action plan for improvements.

Considerations for improvement:

According to Poppendieck and Poppendieck [9] the two ways to reduce cycle time for a process are to achieve a steady arrival rate for work and to look at how the work is processed and remove variability in the processing time.

For this product, we found that the requirements were handed over in large batches and without adequate prioritization.

The perception was that there are a high number of service-level analyses (BUC studies) being done in parallel and it was seen in the data as well. There were as many ongoing and completed service-level analyses that were not even highly prioritized as the ones with a high priority. This was done without taking into consideration the capacity of the organization. Therefore, the current push based approach should be replaced with a pull driven approach that takes into account the capacity of the development organization and the priority of the business opportunity.

One indication of too many ongoing studies is reflected by the dissatisfaction with the quality of information about the breakdown of BOPs to BUCs (see Table 6.3).

Another indication is the sheer overwhelming feeling that the teams have of working with service-level analyses (BUC studies). Yet another quantifiable measure is the fact that none of the BUCs have reached the system-test or the service-test levels in the product development life-cycle.

When seeking explanation for the reasons behind certain challenges, it was noticed that a majority of the challenges were due to the secondary importance given to the documents necessary to ensure the non-functional attributes of the system. For example, in Table 6.4, when asked about the reasons why architectural model and functional baseline were not created in time, the reasons include that the value was underestimated or that there was a time pressure. A similar mindset towards quality was noticed in the release plan where functional requirements were prioritized over non-functional requirements (see Figure 6.6).

Furthermore, the consistent under-estimation (see Figure 6.5a) raised the question, whether the BUCs were broken down to the appropriate scope?

For the future, the key improvements identified were as follows:

- Release frequently and take in smaller batches [9]. “Do not keep piling up work that cannot be used immediately” [9]. This ensures a continuous flow of requirements instead of batches/chunks of business opportunities

- Improve prioritization of business opportunities and BUCs to address the issue of parallelism
- Efficiently deliver new BUCs to the development organization (*pull driven*) also helps to address the issue of parallelism. No need to keep piling up BUCs (through hastened BUC studies) if the development organization does not have the capacity to implement them.

6.5.5 Future state map

Lesson learned: The current process map (see Figures 6.2 and 6.4) did not allow us to analyze the process in sufficient detail to address the identified challenges in the process (see Table 6.3). With these process descriptions, we could have modified the process a bit on a high-level, however, improving it would not have been possible without the awareness of additional information in each step of the process.

With the improvement suggestions (as discussed in Section 6.5.4) and key challenges that were related to information needs (see Section 6.5.3) in consideration, the FLOW methodology was used. Using the high-level list of activities (see Figure 6.4) as input, the FLOW methodology helped to systematically elaborate and visualize the state and storage of information, and also the information and competence required to perform these activities. This facilitated to identify causes for certain issues where the flow of information was broken and to identify concrete improvement actions.

Figure 6.8 shows the outcome of the VSM activity using the FLOW notation. This figure is not intended to explain the development process at Ericsson, rather the intention is to give a flavor of the size and complexity of the process being analyzed. Also, any practitioners applying the proposed approach can see what can be expected as an outcome from it. The improvements in the process are coded in blue color and are listed below:

- The information required to define a business opportunity (BOP) was established
- Prioritized list of BOPs based on the market window by the product management team.
- Acceptance criteria for an initial BUC definition to become sufficient for detailed analysis.
- Involving product managers after initial analysis to validate if the understanding of BOP is correct and the direction of BUC specification is aligned with the expectation in the BOPs.
- Identified what is missing in the detailed BUCs that needs to be added for system teams to perform system level analysis.

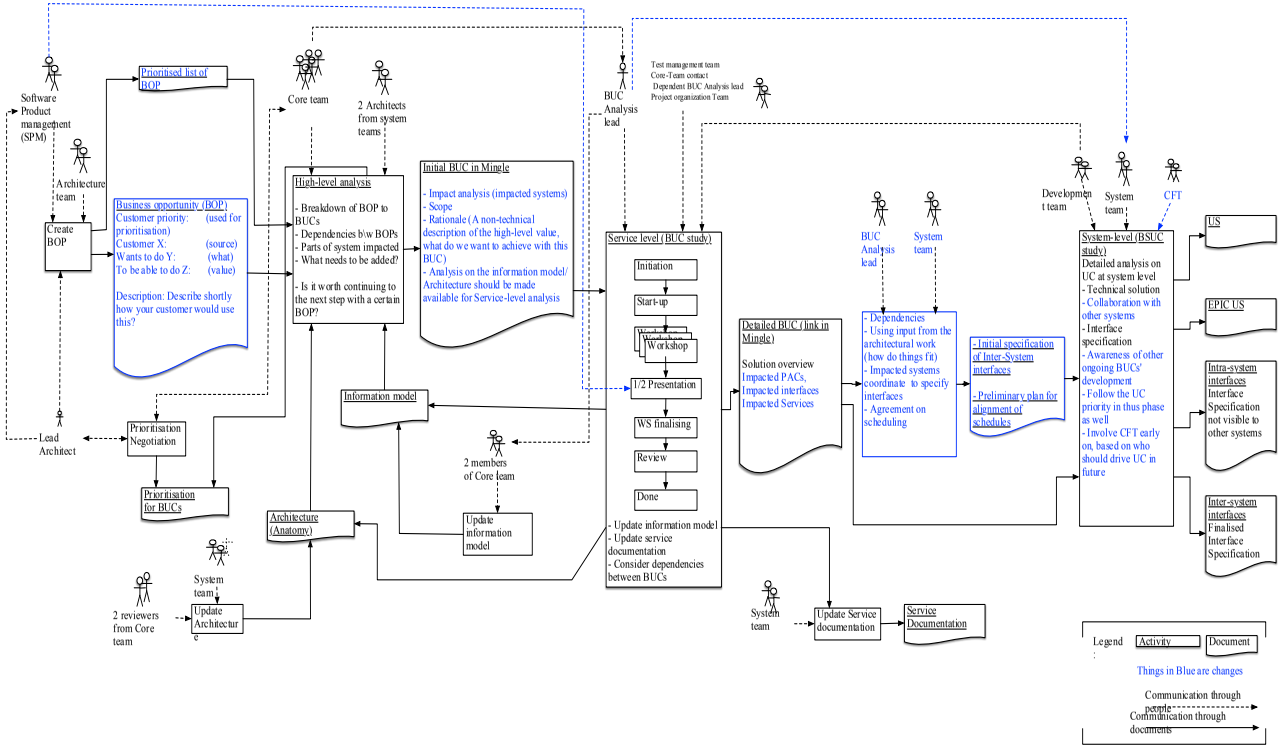


Figure 6.8: Future state map

[redacted]

- Added an activity between service level analysis and system level analysis that should cover the inter-system concerns in more detail for BUCs that require more than one system to deliver. The BUC analysis lead along with the impacted system-teams is required to develop an initial specification of inter-system interfaces and a.preliminary schedule for development.
- It was decided to involve the *Cross Functional Teams* early on in the BSUC study activity.
- Detailed list of tasks that should be performed as part of the system-level analysis (BSUC study) were detailed as well.

Furthermore, with the visualization of overall information flows other challenges that were not visible before were highlighted. For example, it was observed that the prioritized list of BUCs was not available to all the sites and was maintained in a spreadsheet. Similarly, given that the quality of architecture description was considered an issue, it was visible that the resources (only two persons whom are assigned to communicate with all the system teams and to maintain/update the architecture description apart from their other commitments being part of the core-team) were insufficient.

6.5.6 Retrospective

In the last workshop, the practitioners filled out a questionnaire which attempted to assess both the process that was followed to conduct the VSM and also its outcome. Table 6.5 maps the questions to the aspect being evaluated. The questions along with the responses are depicted in Figure 6.9.

Over 60% of the practitioners agreed to that use of FLOW led to more insightful discussions and they would like to see further use of FLOW at the company (see questions Q7 and Q8 in Figure 6.9). While expressing their feedback on the outcome of use of FLOW one practitioner found the notation and the final outcome where the end-to-end process overview, artifacts (key information items), and stakeholders is visible

Table 6.5: The aspects evaluated in the questionnaire.

Question	Evaluated aspect (process/ outcome)	Perspective (VSM/ VSM+FLOW/ FLOW)
Q1, Q2, Q3, Q4	Outcome	VSM + FLOW
Q5, Q6	Process	VSM
Q7, Q8	Process	FLOW

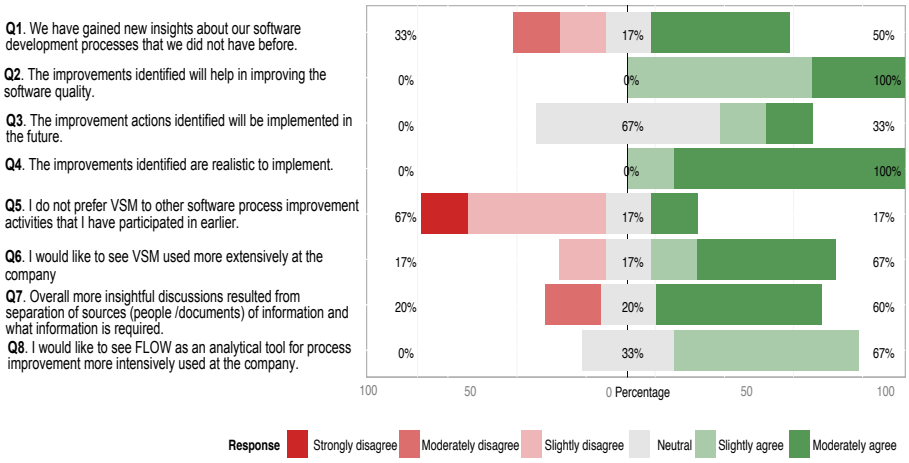


Figure 6.9: Feedback questionnaire results for the six practitioners.

on one sheet is really good *"Best outcome of the exercises"*. Another remarked, *"it is good to depict stakeholder interaction in the process"*.

Similarly, as seen in responses to questions Q5 and Q6 in Figure 6.9, VSM got a majority (67%) of positive feedback whereas only 17% of practitioners disagreed to a more extensive use of VSM in the company. They would prefer other improvement methods over VSM. However, since no information about the other improvement activities that the participants may have been involved in previously was elicited, and due to the anonymity of the survey respondents we cannot explore the negative feedback further.

There were mixed responses to the statement that they gained new insights into their development process (see question Q1 in 6.9). Half of the respondents agreed that they did gain new insights and 33% disagreed. Here it is noteworthy that this is still a positive result, as everyone has to have a good understanding and has to be on the same page for such a large process. Half of the participants were gaining new insights, hence the VSM activity is important to assure an end-to-end flow where everyone knows what information they should provide, or shall receive.

All participants agreed that the improvements identified in the workshops will lead to improved software quality and showed even more confidence in the improvements being realistic to implement (see responses to Q2 and Q4 in Figure 6.9. However, it is

somewhat disappointing to see that only 33% think that they will be implemented and remaining are unsure either way.

One criticism expressed by a participant was that *"the improvements identified were less than what they had expected from the VSM activity"* she also commented that there was a *"high overlap"* during the workshops which were spread overtime.

Two practitioners commented that the workshop highlighted the break in the flow and brought focus to it. Other positive comments highlighted the "granular" level at which analysis was done and how it helped to focus on optimizing towards the overall flow instead of small improvements.

6.6 Experience and reflections

In this study, due to the nature of challenges faced by the practitioners it was realized that most of the challenges are related to information needs and flow in the organization. Although the focus of the study was the specification phase, we wanted to apply the Lean concept of "pull" so that we can identify what information will be required from downstream phases that must be created in the specification phase. Thus, the idea was to make a conscious decision about which analysis to perform, in what detail and whether and how to document it, etc. in the upstream phases thus avoiding any unnecessary analysis and documentation.

Therefore, to "optimize the whole" we started with the release phase of the product and identified what activities are performed during this phase. Then using the FLOW elicitation template the inputs required for those activities were listed and relevant information like the sources and state were elicited. During this workshop, there were two main findings:

- First, we realized that applying the pull principle on the selected scope i.e. starting with the activities in the "release" phase and essentially identifying what needs to be documented and how in the "specification" phase did not work (see Figure 6.2 for an overview of the development process).
- Secondly, we found that eliciting the information and structuring it in the form of FLOW elicitation template (see Figure 6.10) on the fly did not work well in the workshop setting. It was not the template but rather the attempt to structure information while eliciting that become a problem in the workshop setting.

Although conceptually it made sense to identify the information needs in the last activity of the entire development process and then pull that required information from the preceding phase all the way to the start of the development process, but due to the following reasons it did not work in our case: 1) the scale of the product development

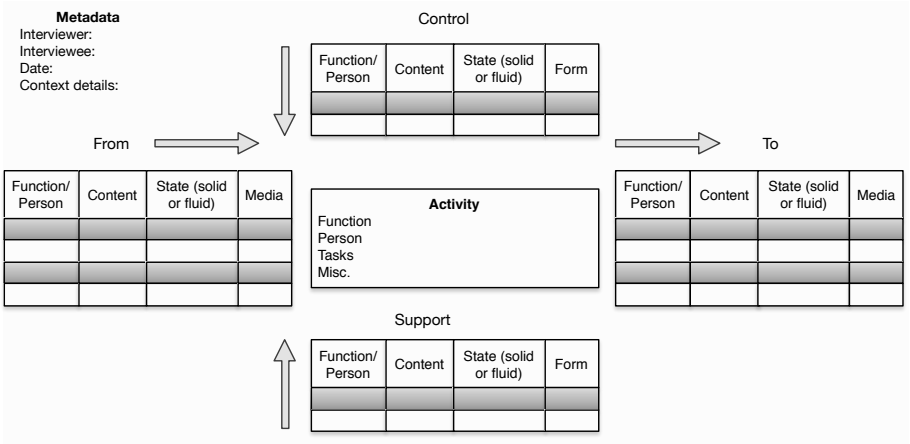


Figure 6.10: FLOW elicitation form (based on [17])

(we lost most of the participants in the discussions, as such only the practitioners responsible for that phase were aware of the detailed tasks, inputs and outputs) 2) because it is a new product still under development and none of the BUCs have reached even the service level testing yet, we realized that soon it became an exercise of assuming what information needs will exist and what and how the required information is assumed to be delivered.

The reason for the second observation we consider is that structuring the information on the white-board in the FLOW template form was not adding value. In a workshop setting, it became very difficult to moderate e.g. some people were now listing outputs and someone mentioned an input and the discussion started on that.

Based on these two observation, we made two changes to our approach, we decided to focus only on the scope of the current VSM i.e. the "specification" phase instead of the end-to-end development process and we used FLOW in the background, guiding the elicitation and to structure and report the findings and resulting process maps (made using the FLOW notation) to the practitioners.

Therefore, taking the activities in the current state map (see Figure 6.4) one by one we elicited the six columns on a white-board as depicted in Figure 6.11. Having elicited this information we used FLOW notation to visualize it. Thus, researchers acted as the interface between flow templates and practitioners thus avoiding the need to train practitioners with the FLOW templates or methodology.

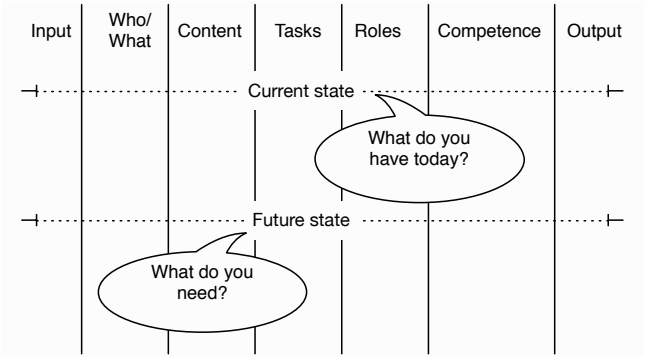


Figure 6.11: Elicitation of information based on FLOW template in a workshop setting.

Figure 6.8 shows the final version of this analysis. The items in blue are what have been identified as improvements in the current process which included: assignment and clarification of roles, establishment of new activities, artifacts and details of their content and the need for competence.

This simple representation provided sufficient support to analyze the process further and identify additional challenges, e.g. deciding about what information should be documented and for which information it is more optimal to rely on face-to-face communication etc.

Compared to the current state map (see Figure 6.4) that was drawn without the use of FLOW methodology, it is visible that these improvements could not have been identified and conceptualized without the use of FLOW.

FLOW methodology was used to guide elicitation and its notation was used to structure information and represent the process using it. The simple notation was easy to adopt for us and practitioners shared the same opinion about the process. It helped us to identify points that needed further clarification and to pinpoint people and artifacts that may be potential bottlenecks in the process.

Key lessons from using FLOW to assist VSM in this case are the following:

- the scope cannot be too broad in such a large product
- the maturity is a factor, and people had a tendency to fall back on the old processes they already know and put them into the future state map (i.e. hard to think on a ”green field”).
- the notation should not be enforced on the participants, this is something that the researchers should do on the background.

- even for a very large-scale process, we could accurately capture the future state map on a level that the practitioners agree and commit to the findings, and this could be represented on a single A3 (i.e. the whole process)
- that companies who have a similar structure (multiple products, need for break-down from abstract to more detailed and then allocating to sub-systems - SoS approach - can benefit from some of the learnings with regard to how to actually structure the process)
- information that was elicited in just a few workshops could most likely not have been gathered by keeping on the abstraction level of the current state map, and that the distinction between information flow types etc. helped to critically appraise the process and identify bottlenecks e.g. the new activity that is added to bridge the information gap between service level analysis and system-level analysis (see Figure 6.8).
- overall, the experience indicates that the approach scales.
- based on the experience from this case, the combination of VSM with FLOW can be useful in cases where the development process is not well defined, if there are challenges related to information exchange, or if the organization is involved in distributed development.

Having diversity in participants that gave a sample of various stakeholder roles from the teams involved in the specification phase of the product development process at the company enabled a systemic end-to-end view of the specification phase and in general the concerns that later phases may have. Their participation ensures that they understand the problems and challenges in other areas that they may not directly be working in. Also, it helps to get a buy-in from stakeholders who may need to do additional work in the future to improve the quality and reduce the lead-time for another process that consumes their output (although it may not reduce their lead-time). In short, it helps to achieve the end-to-end perspective within the specification phase and avoids the pitfalls of sub-optimizing in silos, although we could not reach the same goal for the entire product development process.

Using *"sticky-notes"* to identify challenges ensured participation and gave everyone the opportunity to bring up the challenges they face and consider as a hindrance. Furthermore, by using a variant of the 100 Dollar method for prioritization [1] where each participants has an equal opportunity to highlight what they consider is crucial to address if the organization is to reach the stated goal. In this study, it was seen that although some challenges generated a lot of discussion, but they were not high-up on the prioritized list.

Quantitative analysis, although not a lot of data was available as the process is not well defined and it took considerable effort to extract data from different sites and to we

had to manually connect the data from different silos. Yet the utility became apparent not only to triangulate the qualitative input from the practitioners (e.g. the amount of parallelism in the process) but also to identify new challenges that they had not identified (prioritization of functional requirements over non-functional requirements).

6.6.1 Reflections from the FLOW expert

From the perspective of FLOW, the combination with VSM brought a number of insights: In previous cases, FLOW had been used in a series of interviews. In each interview, one person was asked about his or her concrete activities, and their output and input information. In between two interviews, the graphical FLOW model was updated and extended accordingly. In several cases, flows between two activities were not described consistently from both sides. Other flows were not mentioned at all, leading to "dangling" flows, or obviously unclear flow of information. All those cases triggered focused in-depth questions in the next interviews. Thus, graphical flow models were not shown in interviews, but used in the back office to prepare and focus subsequent interviews. Final FLOW models were only displayed to participants after the series of interviews. There was a summary workshop around the FLOW diagram that had emerged, with a number of observations by the researchers. Activities for improvement were derived in those final discussion workshops. Due to the size and time restrictions, bilateral interviews were not an option in the case described in this chapter. There were too many parts to be discussed and too many people to be involved. Trying to fold a series of interviews into a single workshop (or maybe two workshops) did not work well, as outlined above. In particular, there was no opportunity to update a FLOW model and then reflect on inconsistencies, dangling flows, and the like. The adapted solution used here relaxed the scheme of asking for outgoing, incoming, controlling flows a little bit. Keeping those aspects in mind helped eliciting important information. Doing it in a more informal way than during typical FLOW interviews was advantageous for the workshop setting.

The state map in Figure 6.8 is large and rather complex, but not exceptional for a final FLOW diagram. Initial FLOW models tend to be very simple, covering only one or two activities and their related flows. The models grow with every person interviewed. Figure 6.8 was not updated iteratively after each interview, but drawn in one step after the workshops that served for eliciting information. Once the FLOW state map in Figure 6.8 existed, however, it was used in a very similar way as previous FLOW models in other cases during the final discussion workshop: Participants would point to it, step up to it, correct it, and refer to information flows without necessarily calling them information flows. The diagram helped focus discussions and gain a good overview. Such a large diagram is difficult to draw from scratch, i.e. from the textual notes alone.

However, this adapted variant of information gathering and visualization was obviously more appropriate in this context. This new approach of creating a FLOW model will feed back into the FLOW methodology.

6.7 Follow-up after six-months

The practitioner who acted as the "VSM Manager" during this study conducted a follow-up and reported that almost all the improvements listed in Section 6.5.4 have now been implemented which address the top . Such as:

- "One slider" implemented in BUC study to give overall summary of business needs and impact on sub-systems – Challenge 1 in Table 6.3
- Assigned BUC responsibility for the entire lifecycle – Challenges 6 and 7 in Table 6.3
- To deal with parallelism related problems, a constraint on number of BUCs being developed concurrently has been introduced
- Mechanism to deal with dependency, structure on how to handle cross-PAC dependencies is now in place. Interface descriptions are to be defined before concluding a BUC study – Challenge 3 in Table 6.3
- service and system models and governance for these models have been established – Challenge 4 in Table 6.3
- BSUC study process has been established – Challenge 2 in Table 6.3

Furthermore, having seen the value that VSM analysis can bring they have also collecting data on backlog size and waiting time and it has been connected to the generic measurement dashboard in the company. The process is still not "pull driven" which is a completely different way of thinking and is not intuitively obvious. We are often used to thinking in processing batches. Having said that, the changes that have been implemented lay the groundworks e.g. having the quantitative data on waiting times that will now be collected will help to motivate the transition.

Typically, companies are hesitant to make an investment in change, unless they are fully convinced of the benefits. Since, almost all the improvements identified in this study have been implemented it shows their confidence in the process that was followed in the study and its outcomes.

6.8 Conclusion

This study presents a case study on the usefulness of combining VSM and FLOW. The study was conducted for a large-scale product at Ericsson AB. In cases, such as the one reported here, where the challenges are mostly related to communication and information flows, the use of FLOW methodology will be extremely valuable in providing a systematic approach to identify, visualize and evaluate information flows.

For a new product development, where the processes are less mature and the product is still in its infancy, a typical quantitative analysis of waiting times for artifacts is often not possible and perhaps not as important. The more significant aspect is to establish a process that encourages taking the end-to-end perspective and improving the flow of information and artifacts through the process.

Using VSM assisted by FLOW helped to create a specification process, identify stakeholders and artifacts, and detailed content of these artifacts. This was essential for the processes that consume the output of the specification phase. This approach identified improvement opportunities in the current process and helped visualize them in the future state map.

Another significant benefit of FLOW notation was it simple yet powerful notation that was intuitive and even for a large scale product development process, it could be represented on a single sheet of paper providing an overview of the entire process.

The process for FLOW assisted VSM and its outcome presented in this chapter would guide other researchers and industry practitioners to apply it in their context.

References

[1] P. Berander and P. Jönsson, "Hierarchical cumulative voting (HCV) - prioritization of requirements in hierarchies," *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, no. 6, pp. 819–850, 2006.

[2] B. Berenbach and G. Borotto, "Metrics for model driven requirements development," in *Proceedings of the 28th International Conference on Software Engineering*, ser. ICSE '06. New York, NY, USA: ACM, 2006, pp. 445–451.

[3] A. Cockburn, *Agile software development*. Addison Wesley, 2002.

[4] M. Khurum, K. Petersen, and T. Gorschek, "Extending value stream mapping through waste definition beyond customer perspective," *Journal of Software: Evolution and Process*, vol. 26, no. 12, pp. 1074–1105, 2014.

[5] H. L. McManus, *Product development value stream mapping (PDVSM) manual*, 1st ed., Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 2005.

[6] S. Mujtaba, R. Feldt, and K. Petersen, "Waste and lead time reduction in a software product customization process with value stream maps," in *Proceedings of the 21st Australian Software Engineering Conference (ASWEC)*. IEEE, 2010, pp. 139–148.

[7] K. Petersen and C. Wohlin, "Measuring the flow in lean software development," *Software: Evolution and Process*, vol. 41, no. 9, pp. 975–996, 2011.

[8] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineering*, vol. 13, no. 3, pp. 303–337, 2008.

[9] M. Poppendieck and T. Poppendieck, *Lean software development: an agile toolkit*. Addison-Wesley Professional, 2003.

[10] A. Rausch, C. Bartelt, T. Ternit, and M. Kuhrmann, "The V-Modell XT Applied - Model-Driven and Document-Centric Development," in *Proceedings of the 3rd World Congress for Software Quality*, 2005.

[11] M. Rother and J. Shook, *Learning to see: Value stream mapping to add value and eliminate MUDA*. Brookline, Mass.: Lean Enterprise Institute, Jun. 1999.

[12] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[13] K. Schneider and D. Lübke, "Systematic tailoring of quality techniques," in *Proceedings of the World Congress of Software Quality 2005*, vol. 3, no. 3, 2005.

[14] K. Schneider, K. Stapel, and E. Knauss, "Beyond documents: visualizing informal communication," in *Proceedings of the 3rd International Workshop on Requirements Engineering Visualization (REV '08)*, Barcelona, Spain, Nov 2008.

[15] K. Stapel, E. Knauss, and K. Schneider, "Using FLOW to improve communication of requirements in globally distributed software projects," in *Proceedings of the International Workshop on Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*. IEEE, Aug 2009, pp. 5–14.

[16] K. Stapel and K. Schneider, "Managing knowledge on communication and information flow in global software projects," *Expert Systems*, pp. n/a–n/a, 2012.

[17] ——, "FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW," Fachgebiet Software Engineering, Leibniz Universitt Hannover, Tech. Rep., 2012. [Online]. Available: http://arxiv.org/abs/1202.5919

[18] S. Winkler, "Information flow between requirement artifacts. results of an empirical study," in *Proceedings of the Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2007, pp. 232–246.

[19] A. Wise, "Little-JIL 1.5 language report," Department of Computer Science, Univ. of Massachusetts, Tech. Rep., 2007, accessed: 2015-02-21. [Online]. Available: http://laser.cs.umass.edu/techreports/06-51.pdf

[20] N. Zazworka, K. Stapel, E. Knauss, F. Shull, V. R. Basili, and K. Schneider, "Are developers complying with the process: An XP study," in *Proceedings of the 4th International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. Bolzano-Bozen, Italy: IEEE Computer Society, Sept. 2010, best Paper.

Chapter 7

Identifying and evaluating strategies for study selection in systematic literature studies

Abstract

Study selection, in systematic literature studies (systematic mapping studies and reviews), is a step that is prone to bias and there are no commonly defined strategies of how to reduce the bias and resolve disagreements between researchers. This study aims at identifying strategies for bias reduction and disagreement resolution for systematic literature studies. A review of existing systematic reviews is conducted for study selection strategy identification. In total 13 different strategies have been identified, they were related to achieving/checking objectivity in inclusion/exclusion criteria, providing support in resolving disagreements, and providing decision rules. Furthermore, building on these strategies, a study selection process was formulated and evaluated in a systematic review. The selection process used a more inclusive strategy than the one typically used in secondary studies, which led to additional relevant articles. The results indicate that a good-enough sample could be obtained by following a less inclusive but more efficient strategy, if the articles identified as relevant for the study are a representative sample of the population, and there is a homogeneity of results and quality of the articles.

7.1 Introduction

The goal of evidence-based software engineering is the integration of evidence based on a research question in order to provide the best recommendation to practitioners. This involves different steps that are conducted. The first step is the translation of information needs into a concrete enough question, so that the question could be answered. Thereafter, evidence has to be identified that supports in answering the questions. Not all returned sources of evidence are of equal quality, hence, the evidence has to be objectively selected and evaluated. Thereafter, the recommendation provided by the evidence has to be adapted and implemented (cf. [11]).

Systematic reviews and mapping studies are a commonly used technique in medicine in order to aggregate evidence. Secondary studies [9] [15] are used to explore a variety of topics in software engineering with an aim to answer a research question by conducting an *"exhaustive"* search for relevant literature [9]. Starting with a large set of potentially relevant studies, reviewers rely on several steps of selection, first by reading titles and abstracts, followed by full-text reading for quality assessment [9]. Thus, the reliability of a secondary study is highly dependent on the repeatability of the selection process [17].

Bias in study selection is commonly reported in systematic reviews (see e.g. [14, 12, 16]), and therefore disagreements occur between researchers conducting a review together. In case the titles or abstracts are not clear, the bias is further fortified.

In order to understand the strategies used by researchers to reduce bias and resolve disagreements in systematic reviews and maps a review of existing systematic reviews in software engineering and computer science is conducted. The contribution is a set of strategies that can be followed by researchers. Using the identified strategies, an example combination is proposed and piloted in a systematic review where implications of these strategies on selection results are evaluated. The benefit for research is a common understanding of existing strategies supporting informed decision making of what strategy to follow. Furthermore, it aids in reporting the strategies by making them explicit, provides a structured way to approach the resolution of disagreements and supports an informed decision-making considering the effort spent and the value achieved by different strategies.

The remainder of the chapter is structured as follows: Section 7.2 presents related work. Section 7.3 describes the research method. Section 7.4 explains the identified strategies. In Section 7.5 we formulate a process for study selection. Section 7.6 reflects on the results obtained. Section 7.7 concludes the chapter.

7.2 Related work

In the first guide to conduct systematic reviews [10] different strategies have been proposed. The first strategy is to assess the goodness of the objectivity of inclusion/exclusion criteria by calculating inter-rater agreement using Cohen Kappa statistic. In addition, additional persons should be involved to discuss inclusion and exclusion, especially when the step is done by a single researcher. In case of uncertainty sensitivity analysis is proposed as a solution, but no detailed guide is given of how to conduct the sensitivity analysis.

In the updated guidelines [9] it is added that every agreement/disagreement needs to be resolved through discussion. Another recommendation for single researchers was proposed, namely test-retest. In test-retest a random sample of studies is re-evaluated by a single researcher to determine intra-rater reliability.

Overall, inclusion and exclusion is still a challenge and a common and well defined process for inclusion and exclusion has not been proposed. In fact, an interview study with researchers [2] has shown that study selection and getting agreement are two of the main challenges in systematic reviews. Hence, one of the success factors mentioned is the need of clear criteria.

To the best of our knowledge this is the first study focusing on the investigation of inclusion and exclusion strategies for systematic reviews in software engineering.

7.3 Research method

7.3.1 Research questions

The aim of this study is to identify strategies for reducing bias and resolving disagreement between researchers conducting a systematic review. For this purpose two research questions were asked:

- *RQ1: What strategies are reported within systematic literature reviews in the area of software engineering and computer science?* The first question is answered through a literature review.
- *RQ2: How effective and efficient are identified strategies in including/excluding articles?*

By effectiveness, we refer to the ability to include relevant articles and to avoid their exclusion. Efficiency was measured in terms of the effort (time spent) when using the process. The RQ was answered by using the selection process in a systematic review of software process simulation literature.

7.3.2 Literature review

This section describes the review procedure. We would like to point out that this is not a systematic review aggregating evidence. The goal is to arrive at an overview of what alternative strategies are available, but we are not able to provide information about the accuracy of the strategies.

Study Identification: The study identification was based on three commonly referred to guidelines for systematic reviews (cf. [10, 9] and systematic mapping studies [15]). Articles based on the guidelines have been selected as the guidelines have become a standard reference for systematic reviews in software engineering. Thus, articles that are systematic, but were published before the guidelines have been released, will not be included. The articles citing the guidelines were identified through Google Scholar. Relevant sources for systematic reviews are IEEEXplore, ACM, Elsevier, Springer Lecture Notes of Computer Science, and Wily Inter science. In 2007 Bailey et al. [3] tested different search engines and data bases and found that Google Scholar includes all articles from ACM and IEEE. However, this was not the case for Elsevier. Given the study was done in 2007 and we found articles that are in-print at the publisher (see e.g. [6, 7]) we feel confident that good coverage is provided through Google Scholar.

Study Selection: For the selection of literature the following inclusion criteria were defined:

- The abstract or title has to explicitly state that the article is a literature review or systematic literature review.
- The article is in the area of software engineering or computer science.
- The article is a journal paper, conference paper, thesis, or technical report. As Google Scholar is able to capture gray literature theses and technical reports are considered as well.

Articles are excluded from this study based on the following exclusion criteria:

- Article is not in English.
- The retrieved document is an editorial or an introduction to proceedings.
- The articles is not within the area of software engineering/computer science.
- The article is not accessible in full-text.
- The article is a duplicate of an article already in the set.

The inclusion and exclusion criteria are objective, and are easy to check without requiring interpretation (e.g. looking for the word literature review/systematic literature

review in the title and abstract). Furthermore, there is little risk of bias with regard to the identification of the research area given that the guidelines [10, 9, 15] target software engineering and computer science researchers. As a consequence the choice was made that the first author conducts the inclusion/exclusion process individually.

Data Extraction and Analysis: An article was only considered for data extraction when the review protocol/method section within the article provides information of strategies for reducing bias/resolving disagreements. Such information is usually found under the heading inclusion/exclusion and paper selection, or in the section *"Conducting the review"*. In the data extraction the author names, title, and strategies for each article were extracted. The following process was followed to identify strategies:

- *S1:* Identify the reported strategy and create a code for the strategy. Log the code in the data record for the article currently under review.
- *S2:* Identify the next strategy and determine whether there already exist a code for that strategy. If a code exist, log the code for the article currently being under review, otherwise create a new code and log the code.
- *S3:* Repeat step *S2* until the last article/last strategy in the set has been recorded.

The coding was also done individually by the first author. In the case of strategy identification and documentation of strategies a threat of subjective interpretation is present (see Section 7.3.3).

Figure 7.1 provides an overview of the review process. The guidelines delivered 300 hits for the 2004 version and 122 hits for the 2007 version. The mapping guidelines delivered 19 hits. After applying the inclusion/exclusion criteria, 139 systematic reviews in software engineering/computer science were left. These were used as a basis for strategy identification. Articles that did not report any strategies for bias reduction and disagreement resolution were discarded. In the end of the process 40 articles containing strategies remained.

7.3.3 Validity threats

The three main validity threats in this study are that strategies are missed, bias in the interpretation of strategies, and the issue of the generalizability of the pilot study to other systematic reviews.

Missing Strategies: One threat to validity is that strategies for inclusion and exclusion are missed due to bias of the researcher. This threat is considered relatively low as after reviewing the first 7 of 40 articles 9 out of 13 codes/strategies have been identified and almost all strategies in the remaining articles fit well into these categories. There are two exceptions. Strategy 13 was found in the 35th [1] article reviewed, and strategies 14, 15, and 16 were found in the 41st article reviewed [8]. With each additional

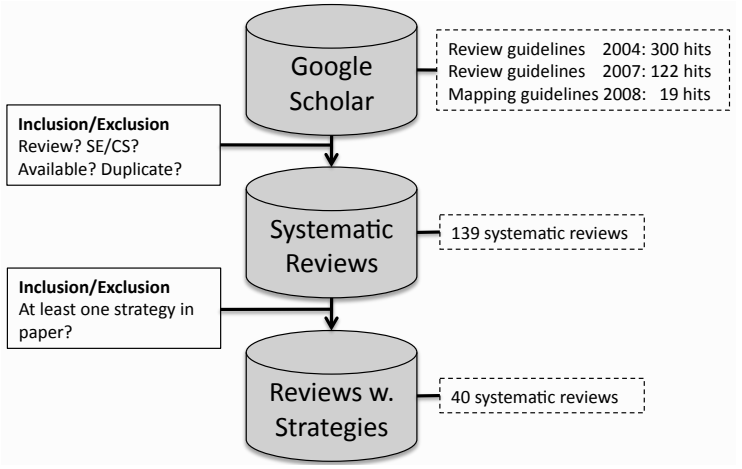


Figure 7.1: Overview of the review process.

review considered the number of newly identified strategies reduced and stabilized after the seventh article.

Interpretation of Strategies: The strategies were interpreted by a single researcher, which makes this step prone to bias.

Generalization of the pilot: The review is focusing on a specific topic area (software process simulation) and aims at only capturing practical application of simulation. Often the abstract is not clear about the actual application of the simulation (i.e. if the application was done in practice or in the lab). Similar problems are likely to occur whenever empirical research is requested as a criterion for inclusion and exclusion (see e.g. [14]). Hence, the inclusion/exclusion situation of the pilot is likely to be generalizable to reviews focusing on empirical studies, or those summarizing the state of practice with respect to real world applications. The selection process has been used in only one systematic review. Thus, further replications are required to derive generalizable conclusions. The analysis in this study is based on the assumption that adaptive reading depth [15] produced as good results as full-text reading.

7.4 Identified strategies (RQ1)

In total, we identified 13 codes representing different strategies. The strategies are grouped according to their goals. Three goals have been identified:

- *Objectivity of Criteria:* Strategies verify the objectivity of the selection criteria.
- *Resolve Uncertainty and Disagreement:* Strategies aid researchers in resolving uncertainties and disagreements.
- *Clear Decision Rules:* Strategies based on decision rules determine whether an article is included or excluded.

In the following an overview of the goals and their related strategies is presented. Each table also contains the references to the studies applying the strategy. It is important to point out that the researchers might have used more of the presented strategies in their studies, but did not report them. However, it is still interesting to observe the number of reported articles as they represent what the researchers think are important strategies when conducting the article selection. At the same time reporting a strategy means that an informed decision has been taken about that strategy.

Table 7.1 presents strategies related to the goal "Objectivity of Criteria". It can be seen that five studies followed strategy O1 and ten studies strategy O2. One possible reason for the frequent usage is that these strategies were recommended in the systematic review guidelines [10, 9]. Objective O1 is related to piloting the inclusion and exclusion criteria. Furthermore, O2 tests the objectivity considering the level of agreement. Objective criteria formulation was explicitly stated as a strategy in [14], the reason being that the review was conducted by a single author.

Table 7.2 presents strategies related to the goal "Resolve Uncertainty and Disagreement". A similar observation as for the previous goal can be made. Both strategies that are frequently reported have been proposed in [10, 9], i.e. to consult additional researchers, and to discuss and resolve uncertainties.

Table 7.1: Strategies to achieve objective criteria.

ID	Code	Description	No. of citations
O1	Objective Criteria Assessment (pre-inclusion/exclusion)	Test sub-sets of articles with two or more persons before starting the actual inclusion/exclusion process, high agreement indicate objective criteria	5
O2	Objective Criteria Assessment (post-inclusion/exclusion)	Reviewers measure their agreement after completing inclusion/exclusion to determine level of objectivity on all or a sample set of studies (can also be done on a sub-set)	10
O3	Objective Formulation of Criteria	Require objective statements (e.g. is X stated, Yes/No)	1

Table 7.2: Strategies to resolve disagreements/uncertainties.

ID	Code	Description	No. of citations
R1	Another person check	Additional reviewer(s) is/are consulted to support in the decision of inclusion or exclusion by reviewing/assessing the result	14
R2	Second vote on uncertain and exclude	Only for articles rated as either uncertain or exclude a second vote is obtained.	1
R3	If disagreement or uncertainty in decision then discuss	If a set of researchers is in disagreement, then a decision for the next step is taken after discussion to resolve the disagreement.	18

Table 7.3 presents strategies related to the goal "Clear Decision Rules". These are new and have not been reported in the guidelines [10, 9]. Strategy D4 is of interest, as this strategy is inclusive leading to more articles in the following steps. At the same time strategy D4 is in line with the recommendation given by Kitchenham [10, 9], which recommend being inclusive in study selection. The remaining strategies were reported in individual studies. It should also be observed that reporting a single strategy does not mean that this is the only strategy used. For example, regarding D4 we can not be sure about the articles where one reviewer considers the article "relevant", and the other considers it "irrelevant". This ambiguity, further strengthens the claim that making strategies explicit is a prerequisite for complete reporting.

Overall, the analysis shows that the strategies that have been mentioned in the guidelines are most frequently reported. However, researchers apply strategies beyond that. In total, nine strategies have been identified that are not part of the guidelines.

Combinations of strategies have been used as well, which is not visible from the tables right away. Most commonly two strategies have been reported. The bubble plot in Figure 7.2 illustrates the frequencies of pairwise combinations of strategies. Only four studies reported the usage of more than two strategies (cf. [13, 1, 4, 8]). Figure 7.2 reveals that the strategies proposed in the guidelines are often followed together (e.g. R3 with O2, and R1 with R3) [10, 9].

7.5 Evaluating study selection strategies (RQ2)

The identified strategies from literature can be summarized as the following three main sets:

Table 7.3: Rules to directly arrive at a decision.

ID	Code	Description	No. of citations
D1	Majority Vote	The group of researchers take a vote on the article and the decision of the majority is followed	1
D2	At least one "relevant" then include	If one of the reviewers considers an article relevant then it is included as a primary study	1
D3	All "relevant" then include	Only if all reviewers consider the article relevant then it is included, otherwise it is excluded	1
D4	At least one "uncertain" then include	If one of the reviewers is uncertain regarding the article, it is considered in the next step of the systematic review	11
D5	One "irrelevant" and one "uncertain" then exclude	If one of the reviewers says the article is irrelevant and the other is uncertain then the article is excluded	1
D6	All researchers vote "uncertain" then include	If all researchers are uncertain, then the article is included	1
D7	All "irrelevant" then exclude	If all reviewers agree that the article is irrelevant then it is excluded, otherwise it is included	1

- *Formulate criteria objectively and evaluate objectivity by calculating the agreement level between reviewers*

- *Specify the decision rules to determine whether an article is included, excluded or there needs to be further investigation to decide about the selection result*

- *Specify rules to aid reviewers in resolving uncertainties and disagreements e.g. consulting additional information, seek an additional reviewers' input, voting, or discussion*

Individually these strategies are insufficient to ensure repeatability of the process however none of the existing reviews reported them together. This raised two open questions i.e. which strategies should be combined and what order should they be combined in, to get higher effectiveness and efficiency in the selection process? To provide a partial answer to these more general questions we devised a study selection process (see Section 7.5.1, which builds on results presented in Section 7.4) that was used to answer research question RQ2. The proposed process is presented in Section 7.5.1 and the results of selection are reported in Section 7.5.2.

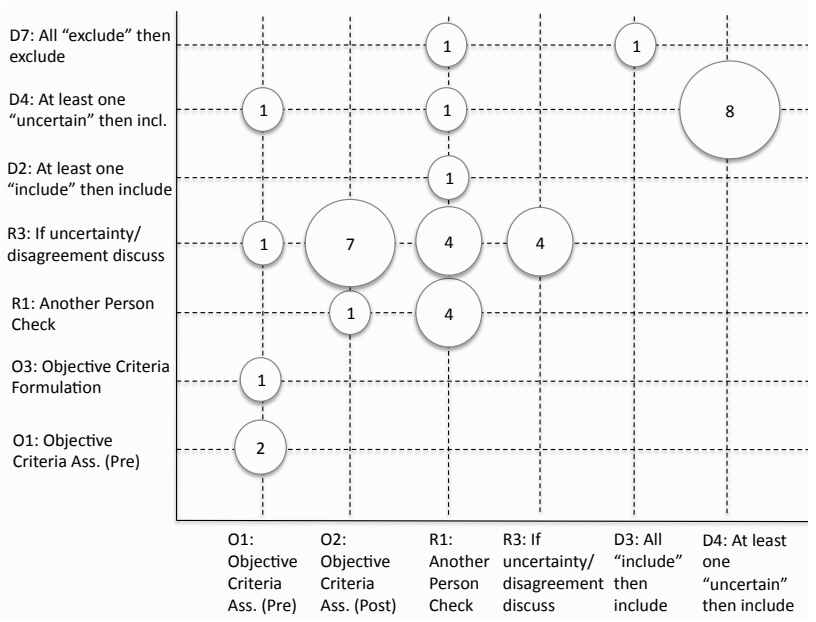


Figure 7.2: Combination of strategies.

7.5.1 Study selection process

An overview of the selection process used in this study is shown in Figure 7.3. To improve the quality of a secondary study, it is recommended that study selection is guided by already decided upon criteria in the study protocol [9]. As a first step, the criteria were specified as questions that can be objectively answered.

Once the criteria have been updated based on the review, the next step is to employ what we referred to as *“think-aloud protocol”*. In this step, reviewers apply the selection criteria on randomly selected articles from the pool of potentially relevant articles. A reviewer, while applying the criteria, also expresses their thoughts and rationales for their choice, which helps to clarify any ambiguities and unintended interpretations. This contributes to improving the internal consistency of studies.

The next step is to pilot the selection criteria [9] on a randomly selected subset of articles independently by the reviewers. Calculating inter-rater agreement values will indicate the objectivity of the criteria. If the results show disagreements, discussing these articles will help to clarify the ambiguities.

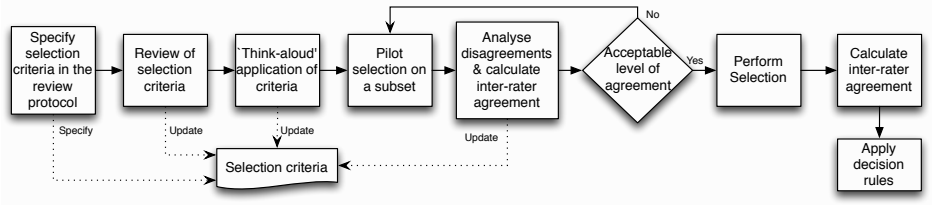


Figure 7.3: Overview of the selection process.

Once the reviewers are satisfied with the level of agreement, we apply the selection criteria on articles independently. Calculating the inter-rater agreement again should show confidence in the repeatability of the study. However, even with an acceptable level of inter-rater agreement we have to consciously decide how disagreements will be handled.

Use of the three possible labels for classifying an article by each reviewer i.e. relevant, irrelevant or uncertain results in a number of possible combinations of disagreements depending on the number of reviewers. The possible outcomes of the selection process with two reviewers with the above three possible classifications for each article are presented in Table 7.4. Unlike the list of possibilities identified from literature (see Section 7.4), this is a comprehensive list of categories.

This structured way of analyzing the disagreements helps to see the level of disagreement and use this knowledge to focus additional effort for further analysis where it is needed. Moreover, the documentation of this systematic approach will improve the repeatability of studies as both the decision choices and the rationales will be available. Existing decision rules used in secondary studies (see Table 7.3) make decisions like "if all reviewers agree that the article should be excluded then it is excluded, otherwise it is included" would mean excluding articles in category *'F'*. However, such decision

Table 7.4: Different scenarios from study selection.

		Reviewer 2		
		Relevant	Uncertain	Irrelevant
Reviewer 1	Relevant	A	B	D
	Uncertain	B	C	E
	Irrelevant	D	E	F

rules ignore the information that is now available about other articles e.g. articles in category *'E'* are most likely irrelevant and should not be taken for full-text reading.

An inclusive strategy will be to review and discuss all the articles in categories *'D'* and *'E'* to re-categorize such articles into categories *'A'*, *'C'* or *'F'*. Articles in category *'A'* and *'B'* should be included for full-text reading given the strong indication of relevance and articles in category F should be excluded.

Articles ending up in the uncertain category *'C'* could potentially be useful, but the title and abstract does not provide sufficient information to make an informed decision. To further investigate these articles in category *'C'*, adaptive reading depth can be used [15]. The outline of three step process used: read the introduction of the article to make a decision; if a decision is not reached read the conclusion of the article; if it is still unclear, search for the keywords and evaluate their usage to describe the context of the study in the article; if a decision is not reached mark the article as uncertain.

Table 7.4 of decision possibilities and their implications can be generalized for more than two reviewers. As an example, category *'A'* is reformulated to *if all reviewers classified an article as "relevant" implies that it should be included*; category *'B'* to *if more reviewers classified an article as "relevant" or "uncertain" than "irrelevant" implies an indication of relevance of the article*; etc.

Each reviewer applies the criteria individually, and logs not only the decision but also what stage of the adaptive reading process the decision was made. This information helps to discuss the disagreements and retrace the decision rationale faster. For inclusiveness, any articles where there is no consensus to exclude or include along with articles marked as uncertain are taken further for full-text reading.

7.5.2 Results of selection

A pilot study of the formulated selection process was conducted in the context of a systematic literature review aimed to aggregate evidence of the usefulness of software process simulation modeling (reported in Chapter 2). The main research question for the systematic literature review was: *What evidence has been reported that software process simulation models achieve their purposes in real-world settings?*

A search string was developed with relevant keywords and used in eight digital databases and one search engine. As the subject is well researched, we had a relatively large set of ***1906*** potentially relevant articles after the initial screening. The large number of search results and involvement of three reviewers (in the selection of studies) provided a good setting to apply the proposed selection process.

The initial protocol which was developed by the first two reviewers and reviewed by the third reviewer. It was decided that while applying the criteria only three possible outcomes will be used by each reviewer: relevant, irrelevant or uncertain (i.e. need

more information than the title and abstract). The criteria were specified very objectively and the level of details of articles consulted and also the role of reviewers was specified.

Think-aloud protocol: After updating the selection criteria/process based on the review, two reviewers applied the selection criteria on five randomly chosen articles together. Reviewers used *"think-aloud protocol"* during selection. This helped to develop a common understanding of the criteria.

Pilot selection: After this step a pilot selection was performed where both reviewers independently applied the selection criteria on a randomly selected subset of 20 articles. Only two articles out of the 20 were classified differently by the two reviewers. These articles were discussed to understand why a difference existed and a consensus on interpretation of the criteria was reached and the protocol was updated. A high agreement upfront indicates that using "think-aloud protocol" helped improve the objectivity and understanding of the criteria.

Selection results: Two reviewers applied the selection criteria individually on the complete set of 1906 articles. The results are shown in Table 7.5. Inter-rater agreement was calculated to assure that the criteria were working well on the overall set of articles (percent agreement: 92.50 and Cohen's Kappa statistic: 0.73) which shows good agreement level.

Out of the 1906 articles on which the selection criteria were applied, 112 articles ended up in categories *'D'* and *'E'*. These articles were discussed and recategorized in one of the other categories *'A'*, *'F'* or *'C'*. The final selection results after discussion are shown in the third column of Table 7.5.

Adaptive reading depth: A pilot application of the procedure for *"adaptive reading depth"* was done on a subset of five articles independently by both reviewers. The results were discussed to develop a shared interpretation. After this, we applied it on all 174 articles that were earlier categorized as *'C'*. 100 articles where both reviewers

Table 7.5: Results of applying the selection criteria

Cat. ID	Number of articles	Number of articles post-discussion	Remarks
A	96	106	Accepted for full-text reading
B	34	34	Accepted for full-text reading
C	122	174	Adaptive reading depth
D	30	0	Recategorized after discussion
E	82	0	Recategorized after discussion
F	1542	1592	Excluded

marked them as irrelevant were excluded and the remaining 74 articles were included for full-text reading (with percent agreement: 78.60 and Cohen's Kappa statistic: 0.53 showing moderate agreement). So in total we found 214 potential primary studies that were read in full-text. From an effort perspective, the adaptive reading took a maximum of 15 minutes per article.

Results of decision rules: From the 214 studies identified as potentially relevant articles, only 87 were included as primary studies. With this final list of primary studies, we can reflect on the decision choices during the selection process and loss/added value in terms of potentially relevant articles as shown in Table 7.6. Here ***"Overhead"*** is defined as the percentage of irrelevant articles that had to be analyzed to identify relevant articles from a certain set of articles. ***"Contribution"*** is defined as the percentage of articles (from this set that became the primary studies in the review) of the total articles[1].

Of the seven decision rules identified in existing secondary studies (see Table 7.3), only decision rule D1:*''Majority vote i.e. a group of reviewers take a vote on the article and the decision of the majority is followed"* could not be evaluated in this study as only two reviewers applied the selection criteria. The rest of the six decisions, each can be seen as a combination of categories *'A'-'F'* from Table 7.4. The decision rules (see Table 7.3) are mapped to our classification of disagreements in the first column of Table 7.7.

These decision rules can be analyzed for their effectiveness in identifying relevant primary studies as shown in Table 7.7. It can be seen that the existing rules vary from the most inclusive strategy (D7:$\{A+B+C+D+E\}$) to the least (D3:$\{A\}$ which identifies less than 60% of primary studies). Thus, following *D3* and *D2* would have meant loss

[1]For *A* (using values in Table 7.6), overhead is (100*(96-51)/96) and contribution is 100*51/87 where 87 is the total number of primary studies.

Table 7.6: Decisions categories and contributions.

Category ID	Number of articles	Of which primary studies	Overhead	Contribution
A	96	51	46%	59%
B	34	11	67%	13%
C	122	16	86%	19%
D	30	3	90%	4%
E	82	6	92%	7%

Table 7.7: Impact of strategies on selection results.

Strategy	Number of articles	Of which primary studies	Overhead	Contribution
D3:{*A*}	96	51	46%	59%
D2:{*A*+*B*+*D*}	160	65	59%	75%
D6:{*A*+*B*+*C*}	252	78	69%	90%
D5:{*A*+*B*+*C*+*D*}	282	81	71%	94%
D4:{*A*+*B*+*C*+*E*}	334	84	74%	97%
D7:{*A*+*B*+*C*+*D*+*E*}	364	87	76%	100%

of 41% and 25% respectively, which could mean a significant threat to the reliability of the secondary study.

The most commonly used decision rule *D4* in existing secondary studies (see Table 7.3) is not the most inclusive strategy. This strategy includes articles in category *'E'* which have a strong indication of being irrelevant and yet the articles in category *'D'* are excluded where one author has marked the article as relevant and other has marked it as irrelevant. Such articles could be just mistakes or may point out serious problems in the selection criteria used to guide the reviewers. Therefore, such articles should not be excluded without reflection.

7.6 Discussion

This section reports on the lessons learned from the literature review for strategy identification and the pilot study.

Lesson 1: Assure repeatability by making strategies explicit and by facilitating adoption: One aim of the systematic reviews is to follow a repeatable process [9]. Therefore, it is imperative that the selection criteria and steps to resolve disagreements are documented and reported in systematic reviews. Doing this will also show that a conscious decision was made from the different available choices and bring more transparency in the review process. Furthermore, the availability of guidelines leads to adoption and reporting, as in the strategy identification review it was clearly visible that the guidelines reported in [10, 9] were the ones most frequently adopted. It was also noticed that the decisions rules in the reviews did not clearly explain how the various possibilities were handled, e.g. 11 studies use *D4* which states to include an article if there is at least one "uncertain" (as shown in Table 7.3). It is not clear whether it means we include the studies even if two of the three reviewers classified it as irrelevant. If

it does, is it a justified choice considering the effort that will be put again to review something that might very well be irrelevant.

Lesson 2: Use and interpret the inter-rater statistics with care: Large number of obviously unrelated articles found in automatic searches can skew the reviewers agreement levels (if measured by the percent agreement) and give false confidence in the objectivity of the criteria. Furthermore, often once the reviewers have an acceptable level of agreement they divide the remaining articles in disjoint sets to reviewers (cf. [5]). As shown by the results here a high agreement in the pilot does not mean that the overlap of articles between reviewers can be avoided. For example, in the given case, up to 20 articles (see contribution of *'B'*, *'D'* and *'E'* in Table 7.6) could have been lost if two reviewers were not involved.

Lesson 3: Always pilot the inclusion/exclusion criteria: The pilot selection during the review is very important as it gives a shared understanding of the criteria and makes the reviewers more consistent. Furthermore, we found employing the *"think out aloud protocol"* very helpful in developing a common interpretation and application of the criteria. From previous experience with systematic reviews we know that rework in selection criteria could lead to much additional effort. Another possible strategy for single reviewers conducting the pilot is to start off with preliminary criteria, and refine them by sequentially reviewing the articles. When the criteria stabilized (e.g. after 20 articles) they can be distributed to the review team.

Lesson 4: Adaptive reading is cost-effective when following an inclusive strategy: The decision to use adaptive reading on articles where there was a disagreement between the reviewers or the available information was the reason for indecisiveness proved a cost effective technique, as no full-text review is required for borderline articles. As with some extra effort we managed to include the articles which would have otherwise been rejected. Therefore, the extra effort in adaptive reading is justified by the number of articles found relevant out of the total articles requiring review.

Lesson 5: Level of disagreement indicates potential for finding relevant articles: Existing rules do not take this into consideration e.g. decision rule *D4* takes any article with a conflict further for full-text reading while *D3* excludes any articles where the reviewers have a difference of opinion. In this study, we took a unique approach to categorize the articles in categories demanding different levels of attention from the reviewers based on the type of disagreement (as shown in Table 7.4) and their potential for identifying primary studies. This indication was corroborated by the results of the systematic review as shown in Table 7.6, e.g. categories *'A'*, *'B'* and *'C'* where neither reviewer had considered the article as irrelevant, contributed to identifying almost 90% of the primary studies. Similarly, articles in categories *'D'* and *'E'* where at least one author considered an article irrelevant only contributed in slightly over 10% of the primary studies.

Lesson 6: Following the most inclusive strategy leads to overhead with little gain: In the case of the systematic review under investigation we observed that the most inclusive strategy only provided little gains in terms of relevant articles identified with a considerable overhead, which is visible in Table 7.7.

With further evidence (by analyzing these decision rules in secondary studies on other topics), we may conclude that there is a potential for investing much less time, and still achieving a good sample. Though, this will only be true if the sample on which the selection criteria are being applied is a true representative of the total population (c.f. [17]) and if there are certain consistent trends in results i.e. losing a few relevant articles will not alter the conclusions of the study altogether.

Lesson 7: Less inclusive strategies seem to lead to a loss of articles: The decision rules *D2* and *D3* result in significant loss of relevant studies as shown in Table 7.7. Unless we have more evidence to substantiate Lesson 4, we should avoid following such less inclusive strategies. Pursuing the most inclusive strategy as a general principle, will reduce the likelihood of overlooking relevant articles and thus improve the consistency [17] of secondary studies.

Similarly, in the given case, there is a very small overhead of approximately 2% in moving from the most commonly used strategy *D4* to the most inclusive decision strategy *D7*. The extra articles that have to be reviewed in this strategy can be handled in a cost effective way by using adaptive reading depth [15]. As for articles with uncertainty and borderline cases the full-text reading is unnecessary.

7.7 Conclusion

This study targets specific problems of selection criteria and process for resolving disagreements in systematic reviews and mapping studies. We have explored the existing strategies and rules employed by researchers conducting such studies through a systematic review. Furthermore, we conducted a pilot review with focused research questions, defined selection criteria, rules to resolve disagreement and a process to guide this activity. Some guidelines and important lessons regarding the development and reporting of the selection process are also presented. This study aims to improve the quality of conducting and reporting of secondary studies in evidence-based software engineering.

RQ1: What strategies are reported within systematic literature reviews in the area of software engineering and computer science? Thirteen different strategies for inclusion and exclusion have been identified. Three are used to assure objective inclusion/exclusion criteria to reduce bias, three to resolve disagreements and uncertainties due to bias, and seven defined decision rules on how to act based on disagreements/agreements. We also found that the strategies proposed in systematic review

guidelines are most frequently reported. Hence, it is important to have explicit and defined guidelines for inclusion and exclusion, which motivated the proposal of a well defined process for study selection.

RQ2: How effective (selecting relevant article and excluding irrelevant ones) and efficient (effort required) is the proposed process in including/excluding articles, considering alternative strategies identified in RQ1? The most effective strategy is the most inclusive strategy, as all less inclusive strategies lead to loss of relevant articles. From an efficiency point of view adaptive reading allows to follow the most inclusive strategy with relatively little effort. However, further analysis and studies are required to establish the impact of this inclusiveness on the conclusions of the studies. For example, would the conclusions be different if *D6* was used instead of *D7*.

Given the results, the combination of selection strategies used in this study is a good candidate-solution to become widely adopted for making study selection in secondary studies more repeatable. It supports an inclusive strategy, which should be preferred over a less inclusive strategy. Furthermore, adaptive reading depth makes the inclusive strategy cost-efficient. In the future, we would like to replicate this analysis on other secondary studies and deduce if general rules about the required level of inclusiveness can be discerned.

References

[1] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Information and Software Technology*, vol. 51, no. 6, pp. 957–976, 2009.

[2] M. A. Babar and H. Zhang, "Systematic literature reviews in software engineering: Preliminary results from interviews with researchers," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, 2009, pp. 346–355.

[3] J. Bailey, C. Zhang, D. Budgen, M. Turner, and S. Charters, "Search engine overlaps: Do they agree or disagree?" in *Proceedings of the 2nd International Workshop on Realising Evidence-Based Software Engineering (REBSE 2007)*, 2007.

[4] N. Condori-Fernández, M. Daneva, K. Sikkel, R. Wieringa, Ó. D. Tubío, and O. Pastor, "A systematic mapping study on empirical evaluation of software requirements specifications techniques," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, 2009, pp. 502–505.

[5] H. Edison, N. B. Ali, and R. Torkar, "Towards innovation measurement in the software industry," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1390–1407, 2013.

[6] M. Höst and A. Oručević-Alagić, "A systematic review of research on open source software in commercial software product development," *Information and Software Technology*, vol. 53, no. 6, pp. 616–624, 2011.

[7] S. Heckman and L. Williams, "A systematic literature review of actionable alert identification techniques for automated static code analysis, in print," *Information and Software Technology*, vol. 53, no. 4, pp. 363–387, 2011.

[8] S. Jalali and C. Wohlin, "Agile practices in global software engineering - a systematic map," in *Proceedings of the International Conference on Global Software Engineering (ICGSE)*, 2010, pp. 45–54.

[9] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Department of Computer Science, Keele University, ST5 5BG, UK, Tech. Rep. EBSE-2007-01, 2007.

[10] ——, "Procedures for performing systematic reviews," Department of Computer Science, Keele University, ST5 5BG, UK, Tech. Rep. TR/SE-0401, 2004.

[11] B. A. Kitchenham, T. Dybå, and M. Jørgensen, "Evidence-based software engineering," in *Proceedings of the 26th International Conference on Software Engineering (ICSE)*, 2004, pp. 273–281.

[12] S. Lane and I. Richardson, "Process models for service-based applications: A systematic literature review," *Information and Software Technology*, vol. 53, no. 5, pp. 424–439, 2010.

[13] A. Meneely, B. Smith, and L. Williams, "Software metrics validation criteria: A systematic literature review," North Carolina State University Department of Computer Science, Raleigh, NC 27695-8206 USA, Tech. Rep. TR-2010-2, 2010.

[14] K. Petersen, "Measuring and predicting software productivity: A systematic map and review," *Information and Software Technology*, vol. 53, no. 4, pp. 317–343, 2011.

[15] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation & Assessment in Software Engineering (EASE 2008)*, 2008, pp. 71–80.

[16] R. Prikladnicki and J. L. N. Audy, "Process models in the practice of distributed software development: A systematic review of the literature," *Information and Software Technology*, vol. 52, no. 8, pp. 779–791, 2010.

[17] C. Wohlin, P. Runeson, P. A. da Mota Silveira Neto, E. Engström, I. do Carmo Machado, and E. S. de Almeida, "On the reliability of mapping studies in software engineering," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2594–2610, 2013.

ABSTRACT

Background: The continued success of Lean thinking beyond manufacturing has led to an increasing interest to utilize it in software engineering (SE). Value Stream Mapping (VSM) had a pivotal role in the operationalization of Lean thinking. However, this has not been recognized in SE adaptations of Lean. Furthermore, there are two main shortcomings in existing adaptations of VSM for an SE context. First, the assessments for the potential of the proposed improvements are based on idealistic assertions. Second, the current VSM notation and methodology are unable to capture the myriad of significant information flows, which in software development go beyond just the schedule information about the flow of a software artifact through a process.

Objective: This thesis seeks to assess Software Process Simulation Modeling (SPSM) as a solution to the first shortcoming of VSM. In this regard, guidelines to perform simulation-based studies in industry are consolidated, and the usefulness of VSM supported with SPSM is evaluated. To overcome the second shortcoming of VSM, a suitable approach for capturing rich information flows in software development is identified and its usefulness to support VSM is evaluated. Overall, an attempt is made to supplement existing guidelines for conducting VSM to overcome its known short comings and support adoption of Lean thinking in SE. The usefulness and scalability of these proposals is evaluated in an industrial setting.

Method: Three literature reviews, one systematic literature review, four industrial case studies, and a case study in an academic context were conducted as part of this research.

Results: Little evidence to substantiate the claims of the usefulness of SPSM was found. Hence, prior to combining it with VSM, we consolidated the guidelines to conduct an SPSM based study and evaluated the use of SPSM in academic and industrial contexts. In education, it was found to be a useful complement to other teaching methods, and in the industry, it triggered useful discussions and was used to challenge practitioners' perceptions about the impact of existing challenges and proposed improvements. The combination of VSM with FLOW (a method and notation to capture information flows, since existing VSM adaptations for SE are insufficient for this purpose) was successful in identifying challenges and improvements related to information needs in the process. Both proposals to support VSM with simulation and FLOW led to identification of waste and improvements (which would not have been possible with conventional VSM), generated more insightful discussions and resulted in more realistic improvements.

Conclusion: This thesis characterizes the context and shows how SPSM was beneficial both in the industrial and academic context. FLOW was found to be a scalable, lightweight supplement to strengthen the information flow analysis in VSM. Through successful industrial application and uptake, this thesis provides evidence of the usefulness of the proposed improvements to the VSM activities.

ISSN 1653-2090

2015:05

ISBN: 978-91-7295-302-4