ABSTRACT

Background: Since the 1950s explicit software process models have been used for planning, executing and controlling software development activities. To overcome the limitation of static models at capturing the inherent dynamism in software development, Software Process Simulation Modelling (SPSM) was introduced in the late 1970s. SPSM has been used to address various challenges, e.g. estimation, planning and process assessment. The simulation models developed over the years have varied in their scope, purpose, approach and the application domain. However, there is a need to aggregate the evidence regarding the usefulness of SPSM for achieving its intended purposes.

Objective: This thesis aims to facilitate adoption of SPSM in industrial practice by exploring two directions. Firstly it aims to establish the usefulness of SPSM for its intended purposes, e.g. for planning, training and as an alternative to study the real world software (industrial and open source) development. Secondly to define and evaluate a process for conducting SPSM studies in industry.

Method: A literature review, two systematic literature reviews (SLR), a case study and an action research study were conducted. A literature review of existing SLRs was done to identify the strategies for selecting studies. The resulting process for study selection was utilized in an SLR to capture and aggregate evidence regarding the usefulness of SPSM. Another SLR was used to identify existing process descriptions of how to conduct an SPSM study. The consolidated process and associated guidelines identified in this review were used in an action research study to develop a simulation model of the testing process in a large telecommunication vendor. The action research was preceded by a case study to understand the testing process at the company.

Results: A study selection process based on the strategies identified from literature was proposed. It was found to systemize selection and to support inclusiveness with reasonable additional effort in an SLR of the SPSM literature. The SPSM studies identified in literature scored poorly on the rigor and relevance criteria and lacked evaluation of SPSM for the intended purposes. Lastly, based on literature, a six-step process to conduct an SPSM study was used to develop a system dynamics model of the testing process for training purposes in the company.

Conclusion: The findings identify two potential directions for facilitating SPSM adoption. First, by learning from other disciplines having done simulation for a longer time. It was evident how similar the consolidated process for conducting an SPSM study was to the process used in simulation in general. Second the existing work on SPSM can at best be classified as strong “proof-of-concept” that SPSM can be useful in the real world software development. Thus, there is a need to evaluate and report the usefulness of SPSM for the intended purposes with scientific rigor.
Towards Guidelines for Conducting Software Process Simulation in Industry

Nauman bin Ali
Towards Guidelines for Conducting Software Process Simulation in Industry

Nauman bin Ali

Licentiate Dissertation in Software Engineering

School of Computing
Blekinge Institute of Technology
SWEDEN
Abstract

**Background:** Since the 1950s explicit software process models have been used for planning, executing and controlling software development activities. To overcome the limitation of static models at capturing the inherent dynamism in software development, Software Process Simulation Modelling (SPSM) was introduced in the late 1970s. SPSM has been used to address various challenges, e.g. estimation, planning and process assessment. The simulation models developed over the years have varied in their scope, purpose and simulation approach. However, there is a need to aggregate the evidence regarding the usefulness of SPSM for achieving its intended purposes.

**Objective:** This thesis aims to facilitate adoption of SPSM in industrial practice by exploring two directions. First it aims to establish the usefulness of SPSM for its intended purposes, e.g. for planning, training and as an alternative to study the real world software (industrial and open source) development. Second to define and evaluate a process to conduct SPSM studies in industry.

**Method:** A literature review, two systematic literature reviews (SLR), a case study and an action research study were conducted. A literature review of existing SLRs was done to identify the strategies for selecting studies. The resulting process for study selection was utilized in an SLR to capture and aggregate evidence regarding the usefulness of SPSM. Another SLR was used to identify existing process descriptions of how to conduct an SPSM study. The consolidated process and associated guidelines identified in this review were used in an action research study to develop a simulation model of the testing process for training purposes in a large telecommunication vendor. The action research was preceded by a case study to understand the testing process at the company.

**Results:** A study selection process based on the strategies identified from literature was proposed. It was found to systemize selection and to support inclusiveness with reasonable additional effort in an SLR of the SPSM literature. The SPSM studies identified in literature scored poorly on the rigor and relevance criteria and lacked evaluation of SPSM for the intended purposes. Lastly, based on literature, a six-step process to conduct an SPSM study was used to develop a System Dynamics model of the testing process for training purposes in the company.

**Conclusion:** The findings identify two potential directions for facilitating SPSM adoption. First, by learning from other disciplines having done simulation for a longer time. It was evident how similar the consolidated process for conducting an SPSM study was to the process used in simulation in general. Second the existing work on SPSM can at best be classified as strong “proof-of-concept” that SPSM can be useful in the real world software development. Thus, there is a need to evaluate and report the usefulness of SPSM for the intended purposes with scientific rigor.
Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Prof. Claes Wohlin and Dr. Kai Petersen. Besides their support, participation and guidance for the thesis, they have been an inspiration to me. I have learned a lot from our discussions and their insightful feedback on my work.

I am also grateful to my friends and colleagues at SERL for creating an enjoyable work environment that is conducive for research. In particular, I would like to thank Kai and Ronald for all the great times on and after work.

I am thankful to my loving wife Binish for her continued support and for being so understanding.

Last but not least, I am grateful to my friends and family in Pakistan, especially my parents and my brother Umair for their encouragement and for always being there for me.

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. is an extension of the conference paper:
Kai Petersen and Nauman bin Ali.

‘Software process simulation: A systematic review on rigor, relevance, and impact’, submitted to a journal, 2013.

Chapter 4. Nauman bin Ali and Kai Petersen.

‘Aggregating software process simulation guidelines: Literature review and action research’, submitted to a special issue of the journal ‘Science of Computer Programming.’
This is an extension of the conference paper:
Nauman bin Ali and Kai Petersen.

Papers that are related to but not included in this thesis.

# Table of Contents

1 Introduction ........................................... 1
   1.1 Overview ............................................ 1
   1.2 Background .......................................... 2
   1.3 Research gaps and contributions ......................... 4
      1.3.1 Research questions ................................. 5
      1.3.2 Research method .................................... 6
      1.3.3 Data analysis ....................................... 10
      1.3.4 Validity threats and limitations .................... 10
   1.4 Overview of studies ...................................... 13
      1.4.1 Study S1: Identifying and evaluating strategies for study selection in systematic literature studies ............. 13
      1.4.2 Study S2: Software process simulation: A systematic review on rigor, relevance and impact .................... 14
      1.4.3 Study S3: Testing highly complex system of systems: An industrial case study ............................... 15
      1.4.4 Study S4: Aggregating software process simulation guidelines: Literature review and action research .................. 16
   1.5 Discussion ............................................. 16
      1.5.1 Simulation as an inexpensive method of inquiry ......... 16
      1.5.2 Simulation as an alternative empirical method .......... 18
      1.5.3 Cost of simulation ................................... 18
      1.5.4 Evidence for usefulness of simulation ................. 19
      1.5.5 Strength of evidence generated by SPSM ................ 21
      1.5.6 Implication on SPSM research and practice ............... 22
      1.5.7 Current SPSM research ............................... 24
   1.6 Conclusions ............................................ 25
   1.7 References ............................................. 27
2 Identifying and evaluating strategies for study selection in systematic literature studies  
2.1 Abstract ........................................ 33  
2.2 Introduction ..................................... 34  
2.3 Related work .................................... 35  
2.4 Methods ........................................... 35  
2.4.1 Research questions ............................ 35  
2.4.2 Literature review ............................... 36  
2.4.3 Validity of the Study ......................... 37  
2.5 Identified strategies (RQ1) ...................... 38  
2.6 Piloting inclusion/exclusion strategy (RQ2) .... 41  
2.6.1 Selection criteria .............................. 43  
2.6.2 Process and Strategies ....................... 43  
2.6.3 Experiences and reflections .................. 46  
2.7 Discussion ....................................... 49  
2.8 Conclusion ....................................... 51  
2.9 References ....................................... 52  

3 Software process simulation: A systematic review on rigor, relevance, and impact  
3.1 Abstract .......................................... 55  
3.2 Introduction ...................................... 56  
3.3 Related work ..................................... 56  
3.4 Research methodology ........................... 62  
3.4.1 Research questions ............................ 63  
3.4.2 Need for review ............................... 63  
3.4.3 Search strategy ............................... 64  
3.4.4 Study selection criteria and procedure ...... 67  
3.4.5 Study quality assessment criteria .......... 72  
3.4.6 Scoring model validity ....................... 77  
3.4.7 Data extraction strategy ...................... 77  
3.4.8 Validity threats ............................... 78  
3.5 Mapping results .................................. 79  
3.5.1 Purpose ....................................... 79  
3.5.2 Scope .......................................... 81  
3.5.3 Simulation Approaches ...................... 82  
3.5.4 Most influential real-world articles ......... 83  
3.5.5 Most influential real-world researchers ...... 85  
3.5.6 Cross analysis purpose and scope .......... 85
## Table of Contents

3.5.7 Cross analysis purpose and simulation approaches .......... 86
3.6 Systematic review results ........................................... 88
  3.6.1 Context of simulation studies .................................. 88
  3.6.2 Verification and Validation of simulation models ............. 88
  3.6.3 Rigor and Relevance of primary studies ....................... 88
  3.6.4 Evaluation of simulation for purpose .......................... 89
3.7 Discussions .......................................................... 89
3.8 Conclusions .......................................................... 94
3.9 References ............................................................ 97

4 Testing highly complex system of systems: An industrial case study 107
  4.1 Abstract ............................................................ 107
  4.2 Introduction ......................................................... 108
  4.3 Related work ......................................................... 109
  4.4 Research method ..................................................... 110
    4.4.1 Purpose and research questions ............................... 110
    4.4.2 Case and context .............................................. 111
    4.4.3 Data collection ............................................... 112
    4.4.4 Data analysis .................................................. 115
    4.4.5 Threats to validity .......................................... 115
  4.5 Results .............................................................. 116
    4.5.1 SoS testing process and practices (RQ1) ...................... 116
    4.5.2 Challenges (RQ2) .............................................. 120
    4.5.3 Improvements (RQ3) .......................................... 123
  4.6 Conclusions and discussion ........................................ 129
  4.7 References .......................................................... 131

5 Aggregating software process simulation guidelines: Literature review and action research 135
  5.1 Abstract ............................................................ 135
  5.2 Introduction ......................................................... 136
  5.3 Related work ......................................................... 138
  5.4 Research methodology .............................................. 139
    5.4.1 Literature review ............................................. 140
    5.4.2 Data analysis .................................................. 140
    5.4.3 Action research ............................................... 141
    5.4.4 Validity threats .............................................. 143
  5.5 Consolidated process for SPSM based on literature sources .... 145
    5.5.1 Problem definition .......................................... 146
Chapter 1

Introduction

1.1 Overview

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 [16] 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 Modelling (SPSM) is proposed as an inexpensive [77] [24] mechanism that attempts to address this challenge by providing a proactive means to assess what will happen before actually committing resources for the change [24]. SPSM was suggested in 1979 by McCall et al. [75] 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 lifecycle to long-term organizational evolution), purpose (including planning, training etc.), and approach (system dynamics, discrete event simulation etc.) [62].

The thesis aims to identify, aggregate and assess the evidence reported in literature to evaluate the usefulness of SPSM in real-world (industrial and open source) software development. Furthermore, it aims to provide a process for conducting an SPSM based study in industry. Using systematic literature reviews, case studies and action research
Chapter 1. Introduction

the thesis makes the following contributions:

- Improvement in methodology for conducting systematic mapping studies and reviews by reducing bias and resolving disagreement between researchers.
- Aggregated evidence of usefulness of simulation in a real-world software development context in relation to strength of evidence.
- Gathered good practices in order to provide practitioners with a process that is based on accumulated knowledge of SPSM.
- Given the limited experience reported from real-world applications of an explicit SPSM process, the consolidated process was complemented by putting it to test in an industrial setting.

The remainder of this chapter is outlined as follows: key terms and concepts for this thesis are discussed in Section 1.2. Section 1.3 identifies the research gap and the contributions of the thesis. It also provides the main research questions and motivates the choice of the research methods utilized to address them. Section 1.4 summarises the results of individual studies and Section 1.5 discusses the findings. Section 1.6 concludes the thesis and its implication on research and practice.

1.2 Background

In the 1960s researchers and practitioners realized that software development was more than just creating effective programming languages [16]. It is a complex endeavour requiring creative effort from people guided by an appropriate organization and procedures [16]. Early descriptions of the software development lifecycle were presented detailing various stages of a software product’s lifetime e.g. Royce model [50] and the Spiral model [7]. The lifecycle models provided the conceptual scheme of how the software process should be carried out [16] [52]. These lifecycle models were the foundation for software processes, which add details to provide practical guidance and control on software development [16] [52]. A software process can be defined as: “the coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product” [16].

A software process model is therefore “any representation” of the software process that captures the relevant aspects of a process for the intended modelling purpose [12]. Typical details in such models are activities to accomplish the process objectives, roles of people, artefacts developed and tools to be used [16]. Like any model, software process models provide abstraction [24] and are used when use of the complete process is
undesirable or impractical [12]. Some of the typical uses of software process models include [16], [12]: Developing a shared process understanding by promoting a representation of the process through a model that can also be used for training newly hired personnel, improving the existing processes and proposing new ones through analysis and validation of process models and by prediction of process behaviour, and simulate if the models are enactable to identify potential improvements.

Most software process models treat software development as a static activity [38]. Therefore, SPSM becomes pertinent because:

- It can represent uncertainty in software development that cannot be captured by analytical models [24].
- Dynamic variables like productivity and defect detection rates are not captured in static models [24] [30].
- Behaviour and decisions taken at one stage affect others in complex and indirect ways that needs to be catered for when analysing the process behaviour [24].
- Manipulation of the actual software process is risky and expensive, whereas SPSM is proposed as an inexpensive way to gain insights in to the actual process [30].

Simulation is the numerical evaluation of a mathematical model that imitates the real-world system behaviour [5] [30]. In case of SPSM, the system being modelled is the software process. A software process simulation model is a computerized model that focuses on one or more of the software development, maintenance or evolution processes especially relevant to the phenomenon of interest [24].

Law [42] classifies simulation models along three dimensions: static vs. dynamic, discrete vs. continuous, and deterministic vs. stochastic simulation models. These dimensions are briefly explained as follows:

- A static simulation model represents a system at a particular time or a system in which time plays no role. On the other hand a dynamic model represents the system evolution over time.
- In discrete simulation, the state of the system changes instantaneously at different points in time. A continuous simulation model is a representation over time, where the state of a system changes continuously.
- A simulation model with no probabilistic components is deterministic. While simulation models with some randomness in them are stochastic.

In SPSM, the choice of simulation approach should depend on the particular goal of the study [30] [24]. In general, continuous simulation (e.g. system dynamics) are considered more suitable for strategic analysis, high-level perspectives and long term
Chapter 1. Introduction

trends etc. [30]. On the other hand discrete simulations can make detailed process analysis, resource utilization and relatively short-term analysis more convenient [30].

All modelling purposes and scopes identified by Kellner et al. [24] have been explored in SPSM research over the years. The purposes studied using SPSM include: planning, process improvement, and training. The simulation models in these studies have ranged from modelling a single phase of the lifecycle to various releases of multiple products [45] [62].

1.3 Research gaps and contributions

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

Gap-1: Lack of an explicit strategy to guide the selection process of articles in systematic literature studies.

Gap-2: Systematic mapping studies exist but there is no aggregation of evidence for usefulness of SPSM for the intended purposes.

Gap-3: Lack of guidelines to choose a process for conducting an SPSM study among a multitude of proposed process descriptions.

Gap-4: No aggregation of best practices for each step of conducting an SPSM study.

Gap-5: Lack of industrial application of a process for conducting an SPSM study.

Gap-1 was identified during the design of the review protocol for a systematic literature review (SLR) to evaluate the usefulness of SPSM. It was observed that there were no selection guidelines, which was a serious threat to the reliability of an SLR results and reduced the ability to replicate an SLR.

Contribution: This thesis contributes to the improvement of the SLR guidelines [26] by proposing and evaluating a systematic approach for the selection of studies in SLRs and mapping studies.

Gap-2 was identified by studying the existing literature. It was observed that there are a large number of 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 [26]. Furthermore, they only scope the research area and do not evaluate the evidence of usefulness in the real-world software development settings.

Contribution: This thesis aggregates the evidence of the usefulness of SPSM in a real-world context in relation to the strength of evidence.
Gap-3, Gap-4 and Gap-5 were identified when the SPSM literature was consulted for a systematic process to guide and support a simulation based study in the case company. It was observed that there are a multitude of proposals mainly depending on the simulation approach, sometimes on the experience of modellers and the size of the organization. There are two set of guidelines for individual steps, but both are only for system dynamics based SPSM studies [23] [4].

Contribution: This thesis reports aggregated good practices in order to provide practitioners with a process that is based on accumulated knowledge of SPSM. 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.

1.3.1 Research questions

The main objective of the thesis is to facilitate the adoption of SPSM in industry. A two pronged approach has been adopted towards this goal: first by evaluating the usefulness of simulation for real-world based on existing research and secondly by providing practitioners with a process to conduct an SPSM based study in industry.

The research questions answered in the thesis are:

RQ-1: How well is the claimed usefulness of SPSM supported by evidence reported in studies conducted in real-world software development setting?

RQ-2: What process guidelines are available to assist practitioners to conduct an SPSM based simulation study?

Figure 1.1 depicts how different studies complement each other to progress towards the main objective. Chapter 2 and Chapter 4 report supplementary studies that enabled the execution of studies S2 and S4 to address research questions posed in the thesis.

Chapter 2 reports a literature review to identify strategies for reduce bias and resolving disagreements between reviewers in secondary literature studies (systematic mapping studies and reviews). The resulting process for selection of articles described in Chapter 2 was used in an SLR of SPSM studies done in real-world software development. Chapter 3 reports the findings of this SLR.

A case study, that was undertaken to understand the testing process of our industrial partner, is presented in Chapter 4. This provided a necessary understanding of the industrial context of the company where an SPSM based study was conducted. Using an SLR, we identified and consolidated the process to conduct an SPSM study. The process, its application in the case company and the lessons acquired are reported in Chapter 5.
Chapter 1. Introduction

Figure 1.1: Overview of the thesis: research questions, methods and chapters.

1.3.2 Research method

Research methods most relevant to empirical software engineering [54] include: controlled experiments [58], case studies [15], survey research [13], ethnographies [48], and action research [34]. This thesis can be described as mixed method research [54]. Different methods were chosen and combined based on their appropriateness to provide the necessary data answering the research questions in individual studies. An overview of research methods applied in various chapters is provided in Table 1.1.

A brief introduction of the research methods applied in the thesis are provided below.
Table 1.1: Mapping of the research methods, the context of the studies and chapters.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systematic literature review</td>
<td>2  X  4  X</td>
</tr>
<tr>
<td>Literature review</td>
<td>3  X  5</td>
</tr>
<tr>
<td>Case study</td>
<td></td>
</tr>
<tr>
<td>Action research</td>
<td></td>
</tr>
<tr>
<td>Industry</td>
<td>X  X</td>
</tr>
</tbody>
</table>

Systematic literature review

A systematic literature review intends to accumulate primary studies aiming to improve the understanding, and to ascertain the validity and reliability of claims [26]. An SLR consists of three major phases [26]:

1. Planning the review. In this phase, the need for the SLR is established, and the review protocol is developed.
2. Conducting the review. In this phase, based on the review protocol, identification of research, selection of studies, data extraction and data synthesis is done.
3. Reporting the review. This phase involves writing a report to effectively communicate the results of the review.

A defined review protocol with an explicit search strategy, inclusion/exclusion criteria, and extraction form to guide what information will be retrieved from primary studies differentiates a systematic review from a conventional literature review [26]. Chapter 2 reports a selection process that contributes in improving the SLR methodology by systematising the inclusion and exclusion of studies. This thesis reports two SLRs:

- Chapter 3 reports the SLR that aims 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 [26] and uses the selection process proposed in Chapter 2.
- Chapter 4 reports an SLR that aims to identify existing process descriptions to conduct an SPSM study in industry. Given that there were mapping studies available on SPSM [45] [62], a protocol driven search was not required, therefore we used forward and backward snowball sampling. This SLR used the guidelines proposed by Webster and Watson [56].
Chapter 1. Introduction

Chapter 2 reports a literature review to identify strategies for reducing bias and resolving disagreement between researchers conducting a systematic review. The review was done with a defined search strategy, explicit selection process and a described data extraction and analysis process. However, it is not an SLR to evaluate the accuracy of the identified strategies rather an attempt to get an overview of what alternative strategies are available.

Case study

Given the nature of software development which is very context dependent [42], 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 [15]. It has a high degree of realism, but that comes at the expense of the level of control that one may achieve in experiments. The credibility of results is achieved by triangulation of data sources, observers, methods and theories [15]. 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 [15]: case study design, preparation of data collection, collecting evidence, data analysis, and reporting.

In order to develop a simulation model of the test process in the company (reported in Chapter 5) it was necessary to develop a better understanding of the test process and the broader context of the company. To achieve this understanding of the test process, a case study was designed and conducted at the case company, it is reported in Chapter 4. In this case study, the case studied was a development site of a large telecommunication vendor. Data source triangulation was used to improve the credibility of the results. To understand the test process, interviews (of practitioners with different roles and responsibilities) were conducted and triangulated with the process documentation (of test levels and details of test process) in the company. Similarly the challenges reported in the interviews were triangulated with the quantitative data acquired from the defect database.

Action research

Action research aims to simultaneously introduce an intervention in a natural setting to improve a problem situation and contribute to scientific knowledge [34]. The researcher actively participates in planning, implementing, monitoring and evaluating the impact of change in the company. McKay and Marshall [34] identify two conceptual cycles. These cycles are interlinked, contingent and represent the researcher’s
problem solving and research interests in the study. The steps in these cycles are shown in Table 1.2.

Table 1.2: Action research cycle.

<table>
<thead>
<tr>
<th>Step</th>
<th>Problem solving interest</th>
<th>Research interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>Problem identification.</td>
<td>Setting research aims.</td>
</tr>
<tr>
<td>B.</td>
<td>Understanding problem context.</td>
<td>Finding relevant literature.</td>
</tr>
<tr>
<td>C.</td>
<td>Planning problem solving activity.</td>
<td>Planning and designing research project.</td>
</tr>
<tr>
<td>D.</td>
<td>Implementing the problem solution.</td>
<td>Conducting the research.</td>
</tr>
<tr>
<td>E.</td>
<td>Monitor in terms of problem solution efficacy.</td>
<td>Monitor in terms of research interest.</td>
</tr>
<tr>
<td>F.</td>
<td>Evaluate the effects of action on problem.</td>
<td>Evaluate effect of intervention in terms of research questions.</td>
</tr>
<tr>
<td>G.</td>
<td>Amend plan if further change is desirable.</td>
<td>Amend plan and design if further explanation and research are required.</td>
</tr>
</tbody>
</table>

Together with the representatives from the case company the following problem was identified. They wanted to illustrate, the benefits of early integration and highlight the consequences of missing a test iteration. It was not possible to demonstrate this with the static process diagrams because of the complexity of the test process. The interest was in developing a simulation model that adequately represented the test process at the case company so that it is relevant and realistic for the developers. Also the ability to show the consequences of various what-if scenarios in terms of the flow of requirements in development and various other stages of testing and release.

Action research was the method of choice as we were developing a simulation model in a real-world setting which involved all the steps identified in Table 1.2. For example, the problem was identified in the company, we conducted a case study to understand the problem context and conducted a systematic 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. The details of the action research study in this thesis are presented in Chapter 5. In this study however only the first iteration was completed where the process of conducting an SPSM based study was evaluated. The next iteration will be to evaluate the improvement achieved by the simulation based training in achieving the learning outcomes of a training.

Both, the case study and the action research study, were conducted at the same
telecommunication vendor. Chapter 4 and Chapter 5 provide details of the industrial context for these studies.

1.3.3 Data analysis

In this thesis grounded theory [16] was not used to develop theories, but the concepts from this methodology were used to structure and analyse the qualitative data throughout this research.

In study S3 (reported in Chapter 4) coding [16] was used to categorise the challenges and their influence on the undesired outcomes of fault-slippage, low maintainability of the test suite and long turnaround time for test suite execution. Similarly, the improvement suggestions were categorised, as relevant to various test levels, to requirements engineering and communication, as reported in the interviews.

Contrary to the grounded theory guidelines for coding, in S2 (reported in Chapter 3) a predetermined classification of SPSM purposes and scopes was used. This was done because an existing classification [24] is well established in the community and no data was found that would require updating this classification.

1.3.4 Validity threats and limitations

Different classifications exist for validity threats of empirical research, e.g. for experimentation [58] and for case studies [15]. The threats to the validity of findings in the thesis are discussed using the classification by Runeson and Höst [15]. It was used as two of the studies in this thesis used case study and action research methodologies.

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 [15].

In S1, the interpretation of identified strategies was done by a single researcher, which makes this step prone to bias.

In S2, two researchers were involved in the selection of studies, data extraction and quality assessment and analysis of the data. Different criteria were piloted and the results are presented in the paper. Any differences in pilots and actual execution of the studies were discussed where 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. To minimize the reviewers bias and dependence of review results on personal judgements,
explicit criteria and procedure were used. This also increased the repeatability of the review.

In the studies S2, S3 and S4, the concept of coding was used from the grounded theory for analysis of qualitative data. Coding in studies S3 and S4 was not guided by any predetermined coding categories to reduce the researchers’ bias on data synthesis. This was an attempt to reduce the researcher’s influence on the categorization, but the categories will inevitably have some dependence on the researcher performing the coding. This threat to validity was reduced to some extent with a review of the coding process and results by a second researcher. For practical reasons only one researcher did the coding. In study S2, however, two researchers were involved in the classification.

Study S3 involved three researchers, two of them were actively engaged in the design, review and execution of the study, thus providing observer triangulation. An explicit case study protocol with detailed documented steps of data collection and analysis increased the reliability of this study.

In S4 the selection of studies, 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 threat to validity of results. However, this threat of bias in selection by having explicit objective criteria.

**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 [15].

In S1, a threat to validity was missing a strategy for inclusion and exclusion due to bias of the researcher. This threat is considered relatively low as after reviewing the first 7 out of 40 papers the number of newly identified strategies reduced and stabilized.

In S2, to minimize the threat of missing literature, databases covering computer science, software engineering and management related literature were used. By using an electronic search (with a search string) the selection bias of researchers was reduced. The selection of articles was done using the guidelines presented in Chapter 2, which aims to reduce selection bias and document how the disagreements were resolved.

Similarly, in S4, to minimize the risk of overlooking some relevant literature the search was based on a published systematic literature review and then enhanced using the Webster and Watson guidelines [56]. The primary studies from existing SLRs were taken as an initial set, which were further supplemented by backward and forward snowballing [56]. Furthermore, there is some empirical evidence that snowballing provides improved results when compared to database search [8] [7].

In S4, two researchers were involved in interviews, presentations, review meetings and compared notes and observations after each meeting. This reduced the threat of
misinterpretation of feedback by the researchers.

In both S3 and S4, instead of relying on a perspective of a single role, multiple practitioners having different roles in the company were interview to avoid any bias. Thus, through triangulation using multiple practitioners from different teams the threat of having a biased perspective of the testing process was reduced.

In S3, the challenges identified in the case study were examined in the development context of the company. However, given the complexity of a real world organization and the fact that confounding factors are always a challenge when studying real world systems, the isolated effect of the development characteristics can not be established, and requires further investigation.

**Construct validity**

The threats to construct validity reflect whether the measures used really represent the intended purpose of investigation [15].

In S4, 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 was because of 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. There is a threat of maturation that the authors acquired more knowledge from literature beyond what is expressed in these guidelines.

Similarly, a majority of the process guidelines consolidated in study S4 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 5) system dynamics was the approach of choice. This means that the usefulness of the consolidated process needs to be evaluated for other simulation approaches.

In S3, both methodological, i.e. interviews and archival data analysis and data source i.e. practitioners and defect database triangulation were used to strengthen the evidence generated in this study. Using the appropriate amount of raw data and through a clear chain of evidence (maintaining traceability between the results, qualitative data and sources), this validity threat was minimized.

In S2, 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 step, but due to page restrictions did not report on the results. However, the principle conclusion would not be different.
External validity

The threats to external validity limit the generalization of the findings of the study outside the studied case [15].

In S4, the consolidated process has only been executed with the specific goal of training, using a system dynamics approach for SPSM in the case company. This limits the generalization of conclusions beyond this specific case and simulation approach. Given this limitation of the study, using the description of the company context and their testing process (in this study) other companies in a similar context are likely to find the results transferable to their context [42].

Similarly in S3, no general claims can be made, but the case in this study represents a typical large scale software based product development situation. Therefore, application of this is likely in similar contexts with respect to system complexity, domain, etc.

In S1, the selection process is used to conduct a systematic review on a specific topic (software process simulation in industry). Often the abstract is not clear about the applied context of the study (i.e. if the application was done in practice or in the lab). Hence, the inclusion/exclusion problem faced in this SLR is likely to be generalizable to reviews focusing on empirical studies, or those summarizing the state of practice with respect to real world applications.

1.4 Overview of studies

As shown in Figure 1.1 each chapter in this thesis reports an individual study. In the following sections an overview of these studies, research methods used and major results and conclusions are presented.

1.4.1 Study S1: Identifying and evaluating strategies for study selection in systematic literature studies

Study S1 targets the selection criteria and the process for resolving disagreements in systematic reviews and mapping studies. The aim is to improve the quality of conducting and reporting of secondary studies in evidence-based software engineering. By analysing the existing SLRs the strategies and rules employed by researchers in conducting such studies 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 are the ones proposed in the systematic review guidelines [26]. 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.

The proposed process for study selection was evaluated in conducting an SLR, reported in Chapter 3, 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 to follow the most inclusive strategy with relatively little additional effort.

1.4.2 Study S2: Software process simulation: A systematic review on rigor, relevance and impact

Given that there is considerable literature on SPSM and that there are a number of secondary studies mapping [45][62], but not aggregating evidence, the need to evaluate and aggregate evidence was identified. This study aims to identify which simulation approaches are useful for what purposes in a given context. The focus of this review was the application of SPSM in real-world settings, while the existing mapping studies did not make such distinction. Therefore, a systematic mapping of literature was done followed by an SLR to assess the evidence they have reported.

The results of the mapping revealed that the scope of the simulation models is restricted to a single phase of the lifecycle or the lifecycle 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. A majority of the studies used simulation for planning, process improvement, training and learning respectively. In comparison, only a few used simulation for control and operational management. The most commonly used simulation approaches are system dynamics and discrete event simulation respectively.

To identify the trends in SPSM research, the most influential authors and articles were identified. Among the top ten articles, system dynamics was most often used followed by discrete event simulation and hybrid simulation indicating a high scientific impact and industrial relevance of these approaches. Similarly, all purposes identified by [24] are represented by the top 10 articles with a majority using it for process improvement and technology adoption.

To assess the evidence of usefulness of simulation for the proposed purpose the rigor and relevance of studies, the simulation 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 were analysed.

Among the primary studies in this review, over 60% of them do not sufficiently describe the context of the study. Similarly, 77% of the studies do not report any validation of the representativeness of the simulation model of the software process being studied and 36% studies do not report any validation of the model’s behaviour.

In terms of quality of primary studies, 89% of the studies scored low on scientific rigor and had low-to-medium industrial relevance. Lastly only 14% of the studies reported some sort of evaluation of the model’s usefulness for the suggested purpose.

1.4.3 Study S3: Testing highly complex system of systems: An industrial case study

A case study was conducted at the company, which aimed to understand the testing process and identify challenges and improvement proposals for dealing with these challenges.

The understanding of the test process acquired through this case study was used to develop a simulation model of the testing process in study S4 (presented in Chapter 5). The challenges identified in the study were classified as general testing challenges and as challenges that are unique and amplified in the development context of the company.

Three sources of information were used for this purpose of understanding the test process: process documentation, defect database and interviews with practitioners. Overall six test levels were identified at the company. These were visible in both the test process documentation and in the interview results. Furthermore, the organization of the testing teams and their responsibilities was roughly organized around these levels as well. Some additional test levels were found in the defect database, however, based on the defect reports from the last two years these levels were not used at all in reporting defects. The understanding of the test process was acquired through the test process documentation and the interviews.

This study highlighted the importance of triangulation e.g. how inconsistencies were found between the defect database and the process documentation when it comes to test levels. Similarly, the importance of qualitative information was highlighted e.g. when identifying the challenges in the company related to testing, interviews and defect data indicated different areas in the software process for improvement.
1.4.4 Study S4: Aggregating software process simulation guidelines: Literature review and action research

To find a systematic process for conducting a simulation based study in industry, the SPSM literature was studied. A number of processes was found that were positioned to target a certain simulation approach, size of the organization, modellers experience etc. Therefore, it is not an obvious choice for a practitioner to know which of these processes to follow given the little background in the area of SPSM. This study addresses this problem by analysing the existing processes and consolidating them. Furthermore, this consolidated process was supplemented with guidelines from SPSM and simulation literature in general. 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. Chapter 5 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 used to develop a system dynamics based simulation model of the test process at the case company. The experience and reflection on using this process are also reported in this chapter.

1.5 Discussion

As defined earlier in Section 1.2, simulation is the numerical evaluation of a mathematical model that imitates the real-world system [5] [30]. In the case of SPSM, the system is the software process. SPSM is put in context as a means of studying a system, using the possibilities identified by Law [42]. With software process as the “system” of interest the possible options to study it are presented in Figure 1.2. “Pilot studies” were added where a change is incorporated on a smaller scale but in the real-world settings. Similarly, “descriptive process models” were added as they are often used as a means to understand the software development process and may use a specialized modelling language and formalism to represent the current way of working.

1.5.1 Simulation as an inexpensive method of inquiry

Compared to experimenting with the actual process SPSM is positioned as an inexpensive mechanism of inquiry [59] [77] [24] [30]. However, in our opinion there are two problems with this premise:

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.
• 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 study the strength of evidence should also be taken into account.

Simulation imitates the behaviour of a system and is used as a problem-solving methodology to deal with real-world problems [5]. However, in the SPSM literature it has two implied purposes which are not delineated:

1. Simulation as a problem-solving and an applied tool for decision support.
2. Simulation as an empirical research method.

Figure 1.2: Ways of studying a software process (adapted from [42] with additions marked with a *).
1.5.2 Simulation as an alternative empirical method

Simulation is motivated as an alternative by citing the high cost and risk of manipulating the actual system [77] [59]. 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” [54]. Yet 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].

In the SLR of SPSM literature (see Chapter 3) only two studies reported some estimate of the effort spent on the simulation based study. This lack of reporting may suggest that researchers consider SPSM a research activity and do not report the cost of doing it. After all, do we report the time it took us to set up the lab for the experiment or to motivate the students to participate in it?

The satisfaction of the SPSM modellers with existing tools for simulation is also an indication of using simulation as a scientific tool of inquiry. Ahmed et al. [31] found that all of the SPSM modellers, in their survey, expressed satisfaction with the existing tools. However, when we look at other disciplines using simulation as a decision support tool, it is apparent that the use by practitioners raises unique requirements on the simulation tools, e.g. integration of the simulation model with the existing decision support system [12] [6].

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.

1.5.3 Cost of simulation

When proposing a new tool or practice in industry it is important to have the following prerequisite information:
• Cost of adoption
• Effectiveness over the existing methods.

Pfahl and Lebsanft [85] 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 [84] only reports the calendar time of three months for knowledge elicitation and modelling and four meetings with the client. Shannon [53] 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 need adaptation. Expecting practitioners to put in 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 [6] 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.

1.5.4 Evidence for 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.

Brooks [9] prophesied that, “Not only are there no silver bullets now in view, the very nature of software makes it unlikely that there will be any—no inventions that will do for software productivity, reliability, and simplicity what electronics, transistors, and large-scale integration did for computer hardware”. However, a survey of SPSM literature suggests that he may have been proven wrong, not only that the “silver bullet” exists but it has been found as well (at least for software engineering challenges).

Here is a brief summary of promised benefits of software process simulation (for details see Chapter 3 for applied SPSM and for overall SPSM literature see [45] and [62]):

• Improved effort and cost estimation
• Improved reliability predictions
• Improved resource allocation
• Risk assessment
Chapter 1. Introduction

- Studying success factors for global software development
- Technology evaluation and adoption
- Training and learning

Christie’s [39] 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 3). The lack of discussion of limitations or threats to the validity of findings from the SPSM studies (see Chapter 3) further highlights this problem.

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 [12]. However, it should be considered that these disciplines have an established tradition of using simulation and enough 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 evaluating a value stream mapping workshop [36] with static process descriptions compared to one facilitated with a dynamic simulation model. Or evaluating the effectiveness of simulation based training compared to a seminar with graphical aids.

Unfortunately, based on the SLR in Chapter 3 and the survey results [31], SPSM modellers 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. [28].

The crisis of credibility in SPSM research is due to a lack of verification, validation and evaluation of models (see Chapter 3 and [11]) and poor quality of reporting (see Chapter 3 and [10]). SPSM may take solace in the fact that it is not alone in this crisis and other disciplines have had such problems as well, e.g. Rahmandad and Sterman [47] reviewed the simulation studies in social sciences and concluded that the articles lack enough details to replicate the studies. Similarly Pawlikowski et al. [41] 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.
1.5.5 Strength of evidence generated by SPSM

Yet 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.

Over the years, the voices for SPSM as an alternative to experimentation and case study research have toned down. Now it is suggested as a complement to other empirical methods. For example, Münch et al. [29] recommend combination of software process simulation with empirical studies. Another example is a methodology proposed by Pfahl and Ruhe [27], which complements system dynamics with existing static modelling methods and quantitative measurement. However, a critical discussion of the limitations of SPSM will help positioning it appropriately as a means to supplement other empirical methods.

Some of the limitations discussed below are general limitations of 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 [39]. In large organizations, the documented process, the actual process and perceived process are different. Thus there is some level of uncertainty brought into the software process simulation modelling. 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 in 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 in the conclusions drawn by such studies. Similarly this raises questions about the validity of simulation results when the models are only calibrated with data from the case company using an existing process model.

Measurement challenges

There are no established physical laws that govern software development and SPSM deals with entities that are difficult to quantify [39]. The confidence in the model depends on verification and validation of the structure and behaviour of the model [39]. Similarly our existing understanding of the causal relations is not at the level of physical laws, e.g. on size and effort [23], and for software metrics validation in general [25]. This brings another layer of uncertainty in the software process simulation models.
Chapter 1. Introduction

Lack of data

Olsen [38] 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 [38]. However, a lack of empirical data in the software industry is a common challenge faced by software process simulation modellers [24]. An often used solution in the existing SPSM research is the use of industrial averages and estimates from analytical models. Given the context driven nature of software development [42] 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 [9]. Ironically, the very analytical models that were criticized for their static nature [24] are being used for calibrating the simulation models. Furthermore, there is inconclusive evidence for usefulness of cross company data, e.g. Kitchenham et al. [27] 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 [54]. The lack of data, therefore, adds another layer of uncertainty in software process simulations.

1.5.6 Implication on SPSM research and practice

With the multitude of uncertainty in SPSM reported in Section 1.5.5, we want to make two statements:

• First, it is inappropriate to suggest simulation as an alternative to other established empirical methods no matter how expensive they are (simulation is not a replacement of experiments or case studies).

• Secondly, given the amount of underlying uncertainty it is imprudent to perform deterministic simulations for accurate predictions. Revisiting the original motivations to choose simulation over analytical models shows that one reason was the ability of simulation to handle uncertainty in software development [24].

In our opinion SPSM may be useful as one of the means to sanity test the solution proposals both in academia and industry. This perspective is shown in Figure 1.3 using Gorschek et al. [18] technology transfer model.

Furthermore, as stated earlier, by delineating the use of simulation for software engineering researchers and practitioners the future research can be directed more effectively. Of course some of the challenges will be shared (e.g. calibration in the dearth of data) but such delineation of usage will identify more concretely what is required
by the practitioners (e.g. usability requirements on the tools) to utilize this potentially useful tool.

In this thesis, by consolidating the existing process descriptions from SPSM literature it was shown that the overall process that the practitioners need to follow is independent of the simulation approach, their experience and size of their organization. Useful guidelines for each step in this process were identified and references to further elaborate on key concepts (essentially from the SPSM literature) are also provided.

By comparing the consolidated process for conducting an SPSM based study with other simulation areas, it was demonstrated that there is much to learn from other areas. Useful references that concur with the guidelines in SPSM literature were also identified. Future research should pursue this direction further and take advantage of the existing experience in other disciplines.
1.5.7 Current SPSM research

McConnel [33], referring to the available software engineering (SE) literature, remarked that pioneers of SE have surveyed the “land” and marked the “trails” the settlers now have to turn the “trails into roads” and develop the rest of “education and accreditation infrastructure”. This notion seems to hold true for software process simulation as well. Since the current work can be seen as a strong “proof-of-concept” of applicability of simulation for SE. Simulation models have been developed using various simulation approaches, with different scopes in relation to the software development lifecycle and for various purposes. However, now the need is to systematize SPSM and establish it as a useful means for practice.

There is an academic discussion on whether software development is an art, science, technology or engineering [33] [40] [21] [11]. Setting that aside, in an industry that is already sceptic of software engineering research [21] [60] [17], proposing new tools and techniques without empirical evaluation is not going to help improve the matters. Thus there is a need to conduct more rigorous evaluations of SPSM for its intended purposes in the real-world software development context.

Having extensively reviewed the literature in SPSM any unassuming reader will be convinced that when David Parnas wrote “stop the numbers game” [39] he was referring to software process simulation. Sadly, all the tactics and their consequences resulting from playing “the numbers game” are evident in the SPSM literature. There is a lot of repetition (sometimes even “copy, paste, disguise” paradigm is at play [39]), overarching theme is that simulation can be used for ..., there is a lack of thorough evaluation, once a substantial model is developed the results of using the same model for various investigation are reported separately, and there are occasions of bespoke research where e.g. the SPSM is suggested to support CMMI but the only connection is made in the concluding remarks of the paper. Furthermore, the tunnel view [55] is ever so present in SPSM research, where research was published “from the clique for the clique” [39].

The problem in the software process community at large identified by Fuggetta [16] and its solution are very apt to improve SPSM research as well. He claimed, “the software process community has redone some of the work accomplished by other communities, without taking advantage of the existing experiences. This insufficient willingness to analyse the results and contributions of other areas has slowed down the rate of innovation. Moreover, we have not taken the opportunity to learn from other researchers’ mistakes”. His suggested way forward is to “heavily invest in finding and evaluating commonalities and similarities, rather than identify differences that often appear to be quite artificial” [16].
1.6 Conclusions

Based on the findings of this thesis the following conclusions are drawn:

**RQ-1: How well is the claimed usefulness of SPSM supported by evidence reported in studies conducted in real-world software development setting?**

1. There is no conclusive evidence to claim the usefulness of SPSM for any of the intended purposes. There is a dire need for empirical evaluation of SPSM in real-world software development settings.
2. Current research lacks both scientific rigor and industrial relevance.
3. Future research should utilize the guidelines to conduct and report SPSM based studies with an emphasis on evaluation of SPSM.

**RQ-2: What process guidelines are available to assist practitioners to conduct an SPSM based simulation study?**

1. 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, modellers experience or organizational size.
2. The consolidation of process proposals for conducting an SPSM based study from SPSM literature resulted in a process that is very similar to the one recommended for simulation based studies in general. This suggests that SPSM and simulation in other disciplines are more similar than previously thought.
3. The consolidated process based on the SPSM literature was successfully used to develop a system dynamics model of the testing process at the case company.

**Main objective: to facilitate the adoption of SPSM in industry**

A comprehensive process for conducting an SPSM based study in industry is presented in this thesis, which reduces confusion in choice of a process caused by the plethora of process prescriptions. It can act as a checklist for advanced simulation modellers as well as facilitate novice modellers for conducting SPSM based studies. The utility of this model was demonstrated successfully in this thesis.

This thesis 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.

Given the strong “proof-of-concept” work in SPSM literature it is likely that SPSM is a tool with potential benefits for both research and practice. However, it should be
seen as such, a tool and not as a research method. Making this distinction will allow
future research to focus on improving SPSM from two likely users’ perspective: the
practitioners and the researchers. From practitioners’ perspective, there is a need to:

1. Present evidence of the usefulness of SPSM for real world software development.
2. Investigate the requirements that practitioners have for SPSM, e.g. usability re-
   quirements on simulation tools.
3. Assess the cost of adopting SPSM in terms of training, resources and required
   infrastructure (e.g. stability in processes, measurement programs).

In this thesis, the need to learn from other disciplines has been placed under the
spotlight. Their methodology, reporting guidelines, best practices and lessons learned
should be studied and evaluated for SPSM. A recent position paper by Birkhölzer [1]
titled “software process simulation is simulation too” aligns well with the findings of
the thesis (as shown by the consolidated process and the overlap in the guidelines from
other disciplines). Lastly, it is strongly recommended that the future SPSM research
should follow reporting guidelines (e.g. [10]), emphasize evaluation of SPSM for in-
tended purposes (e.g. [28]) and strictly follow the guidelines by Wernick and Hall [40]
to “minimise the problems with simulations”.

It is encouraging to see that the SPSM community is also reflecting on the reasons
why the SPSM research trend is slowing down [61] and why it has not been widely
adopted in industry [30]. There is a lot of congruence in the future directions for
research identified in this thesis and the ones proposed in position papers and panel
discussions at the 2012 International Conference on Software and System Process, like
[1] [20]. This thesis however supplements the motivation for these proposed directions
with evidence from the current SPSM literature.
1.7 References


REFERENCES


REFERENCES


REFERENCES


Chapter 2

Identifying and evaluating strategies for study selection in systematic literature studies

2.1 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. Furthermore, a process for inclusion/exclusion is presented and piloted. 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. With regard to the proposed study selection process, we found that the most inclusive strategy was followed with reasonable additional effort to review. The proposed review process is a good candidate solution to become widely adopted to make study selection in systematic reviews repeatable. However, it is inconclusive whether the most inclusive strategy should be preferred over a relatively less inclusive strategy.
Chapter 2. Identifying and evaluating strategies for study selection in systematic literature studies

2.2 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. [14]).

Systematic reviews and mapping studies are a commonly used technique in medicine in order to aggregate evidence. Currently, systematic review/mapping as a method is widely used in software engineering research based on guides developed for that research field [10, 17, 12]. Systematic reviews aim at reducing bias through a well defined definition of the steps included in the systematic review. A well defined definition also enables replication. Systematic reviews are also exhaustive, aiming at getting as close to the population of evidence (studies) available to answer a specific research question. Therefore, it is important to capture relevant studies in the search without publication bias, and not to exclude them during the selection of evidence. The selection of evidence is often done in several steps, starting with reading titles and abstracts, followed by quality assessment [17]. However, given that the inclusion and exclusion is done by humans there is a risk of bias, which might lead to exclusion of relevant studies, or inclusion of irrelevant ones. Bias in study selection is commonly reported in systematic reviews (see e.g. [9, 7, 6]), 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. Based on the identified strategies a new systematic approach for inclusion/exclusion is proposed and piloted on an ongoing systematic review currently conducted by the authors. 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.

The remainder of the chapter is structured as follows: Section 2.3 presents related work. Section 2.4 describes the research method. Section 2.5 explains the identified strategies. In Section 2.6 we propose a process for study selection. Section 2.7 reflects on the results obtained. Section 2.8 concludes the chapter.
2.3 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 [17] 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 [13] 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.

2.4 Methods

2.4.1 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 (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 second question is answered through a pilot study on a systematic literature review on software process simulation.
Chapter 2. Identifying and evaluating strategies for study selection in systematic literature studies

2.4.2 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, 17] and systematic mapping studies [12]). 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 Wiley Interscience. In 2007 Bailey et al. [15] 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. [8, 4]) 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, 17, 12] 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. This 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 2.4.3).

Figure 2.1 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.

### 2.4.3 Validity of the Study

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 [3] article reviewed, and strategies 14, 15, and 16 were found in the 41st article reviewed [1]. With each additional
review considered the number of newly identified strategies reduced and stabilized after article 7.

*Interpretation of Strategies:* The strategies were interpreted by a single researcher, which makes this step prone to bias.

*Generalizability of 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. [9]). 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.

### 2.5 Identified strategies (RQ1)

In total we identified a total of 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 2.1 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, 17]. 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 [9], the reason being that the review was conducted by a single author.

<table>
<thead>
<tr>
<th>ID Code</th>
<th>Description</th>
<th>No. of citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>Objective Criteria Assessment (pre-inclusion/exclusion)</td>
<td>Test sub-sets of articles with two or more persons before starting the actual inclusion/exclusion process, high agreement indicate objective criteria</td>
</tr>
<tr>
<td>O2</td>
<td>Objective Criteria Assessment (post-inclusion/exclusion)</td>
<td>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)</td>
</tr>
<tr>
<td>O3</td>
<td>Objective Formulation of Criteria</td>
<td>Require objective statements (e.g. is X stated, Yes/No)</td>
</tr>
</tbody>
</table>

Table 2.2 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, 17], i.e. to consult additional researchers, and to discuss and resolve uncertainties.
Chapter 2. Identifying and evaluating strategies for study selection in systematic literature studies

Table 2.2: Strategies to resolve disagreements/uncertainties.

<table>
<thead>
<tr>
<th>ID</th>
<th>Code</th>
<th>Description</th>
<th>No. of citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Another person</td>
<td>Additional reviewer(s) is/are consulted to support in the decision of inclusion or exclusion by reviewing/assessing the result</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>check</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Second vote on</td>
<td>Only for articles rated as either uncertain or exclude a second vote is obtained.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>uncertain and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>If disagreement</td>
<td>If a set of researchers is in disagreement, then a decision for the next step is taken after discussion to resolve the disagreement.</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>or uncertainty in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>decision then</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>discuss</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3 presents strategies related to the goal “Clear Decision Rules”. The decision rules are new and have not been reported in the guidelines [10, 17]. Strategy D5 is of interest, as this strategy is inclusive leading to more articles in the following steps. At the same time strategy D5 is in line with the recommendation given by Kitchenham [10, 17], as these recommend to be 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 applied. For example, regarding D4 we can not be sure if the article is also included if one reviewer says “exclude”, and the other “uncertain”. This 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 2.2 illustrates the frequencies of pairwise combinations of strategies. Only four studies reported the usage of more than two strategies (cf. [5, 3, 2, 1]). Figure 2.2 reveals that the strategies proposed in the guidelines are often followed together (e.g. R3 with O2, and R1 with R3) [10, 17].

It is also visible that many different strategies have not been reported together at all. This raises two open questions, namely:

- **Question 1:** Which strategies to combine to get high effectiveness in selection (selecting articles relevant for the population and not excluding relevant articles) and efficiency (reduce the effort in article selection and subsequent review steps)?
Table 2.3: Rules to directly arrive at a decision.

<table>
<thead>
<tr>
<th>ID</th>
<th>Code</th>
<th>Description</th>
<th>No. of citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Majority Vote</td>
<td>The group of researchers take a vote on the article and the decision of the majority is followed</td>
<td>1</td>
</tr>
<tr>
<td>D2</td>
<td>At least one “include” then include</td>
<td>If one of the reviewers includes the article then it is considered for being an included primary study</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>All “include” then include</td>
<td>Only if all reviewers include the article then it is included, otherwise it is excluded</td>
<td>1</td>
</tr>
<tr>
<td>D4</td>
<td>At least one “uncertain” then include</td>
<td>If one of the reviewers is uncertain regarding the article, it is considered in the next step of the systematic review</td>
<td>11</td>
</tr>
<tr>
<td>D5</td>
<td>One “exclude” and one “uncertain” then exclude</td>
<td>If one of the reviewers says exclude the other uncertain then the article is excluded</td>
<td>1</td>
</tr>
<tr>
<td>D6</td>
<td>All researchers vote “uncertain” then include</td>
<td>If all researchers vote uncertain, then the article is included</td>
<td>1</td>
</tr>
<tr>
<td>D7</td>
<td>All “exclude” then exclude</td>
<td>If all reviewers agree that the article should be excluded then it is excluded, otherwise it is included</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Question 2:** Which process should be followed (order in which strategies are executed), e.g. at which stage of the process should we follow decision rules, discuss, or calculate inter-rater agreement to achieve effectiveness and efficiency?

In order to provide a partial answer to these more general questions we propose a study selection process and report our experiences in order to answer our second research question, i.e. *How effective and efficient is the proposed process in including/excluding articles, considering alternative strategies identified in RQ1?*

The proposed process is presented and experiences are reported in the following section.

### 2.6 Piloting inclusion/exclusion strategy (RQ2)

A pilot study of our proposed inclusion/exclusion process was conducted in the context of a systematic mapping and review study. It aimed to aggregate evidence of the use-
fulness of software process simulation modelling in real world software development.

The main research questions for the pilot study (systematic mapping and review) were:

- **RQ 1**: For which purposes has SPSM been used in a real-world setting?
- **RQ 2**: Which are the most influential real-world articles on simulation?
- **RQ 3**: Who are the most influential researchers on applying simulation in the real-world?
- **RQ 4**: What evidence has been reported that the simulation models achieve their purposes in the real-world settings?

For comprehensiveness a detailed list of relevant keywords was developed. Using these keywords a search string was adapted and used for search in eight different digital databases and one search engine covering computer science, software engineering and management literature.

The inclusiveness in list of keywords and search databases resulted in a large number of search results. Thus the large number of search results and involvement of
multiple reviewers (in the selection of studies) provided a good setting as a test bed of the strategies and process identified in this study.

2.6.1 Selection criteria

All duplicates, non-English or non peer reviewed articles were excluded from the search results. The remaining papers were subjected to the following criteria for selection:

- Include an article related to simulation of 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 domain were included in this review.

2.6.2 Process and Strategies

An overview of the proposed inclusion/exclusion process and related strategies is shown in Figure 2.3. We take as input the set of articles for inclusion/exclusion process, after removing duplications.

Pilot selection: As was pointed out in [10, 17] piloting and iteration is important to achieve a stable review protocol. After an initial discussion of the meaning of the inclusion/exclusion criteria both reviewers applied the selection criteria on five randomly chosen articles together. Reviewers used “think out aloud protocol” during selection. This helped to develop a common understanding of the criteria. After this step a pilot selection was performed where both reviewers independently applied the selection criteria on a randomly selected subset of 20 articles. The disagreements were discussed to understand why a difference exists and a consensus on interpretation of the criteria was reached. The pilot selection maps to the identified strategies $O1$ (Objective Criteria Assessment (pre-inclusion/exclusion). The high agreement in the pilot selection provided further confidence in the objectivity of the criteria to commence with actual selection of studies.

Objective criteria assessment: Both reviewers applied the selection criteria individually on the complete set of 1649 articles. It was decided that while applying the criteria only three possible outcomes were used by each reviewer: relevant, irrelevant or uncertain (i.e. need more information than the title and abstract). After completing the inclusion/exclusion we evaluated our inclusion/exclusion criteria by calculating the
inter-rater agreement to assure that the criteria were working well on the overall set of articles. This step relates to strategy $O2$.

Handling disagreement: The possible outcomes of the selection process with two reviewers with three possible answers for each article are presented in Table 2.4. We evaluated different choices here to judge which strategy is more likely to discard irrelevant articles while reducing the threat of missing out relevant one. With this motivation the chosen strategy had to be more inclusive. The decisions for articles in categories F and A are obvious:

- **Category F**: As both reviewers are certain about its irrelevance an article can be excluded.
- **Category A**: As both reviewers are certain about its relevance an article can be
included.

Table 2.4: Different scenarios from study selection.

<table>
<thead>
<tr>
<th>Reviewer 1</th>
<th>Relevant</th>
<th>Uncertain</th>
<th>Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Uncertain</td>
<td></td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Irrelevant</td>
<td></td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

The remaining outcome combinations allow to make different decisions with respect to inclusion/exclusion.

- **Category B**: In this case, the inclusive strategy is to treat the article in the same way as if both authors would have said include like in category A. It would not be reasonable to exclude the article without further investigation.

- **Category C**: When both reviewers are in doubt (category C) there is no point of discussing these and we seek more information to make a decision. The most exclusive strategy would be to remove the articles from the set. As we aimed for being inclusive all borderline articles in the study were kept for further assessment.

- **Category D**: This category contains articles with a clear disagreement situation where one reviewer says include and other exclude. The most inclusive strategy would be to simply take the article into the full-text review. The most exclusive strategy would be to simply remove the article. In our case we had a discussion to resolve such conflicts, which led to re-categorization into categories A, C or F (this relates to strategy R3). That is, the discussion concludes that we are either uncertain, include the article, or exclude the article.

- **Category E**: This category indicates that a study is potentially irrelevant (one reviewer is uncertain), while the other reviewers said exclude. However, the inclusive strategy is to investigate the article further, in our case we discussed the article and re-categorized based on the discussion (this relates to strategy R3). The exclusive strategy would be to simply discard the article.
Chapter 2. Identifying and evaluating strategies for study selection in systematic literature studies

The following is a summary of the selection decisions in the pilot study. Articles in categories A and B are included for full-text review, and articles in category F are excluded right away. Articles in category D and E are discussed, and re-categorized. Articles ending up in the uncertain category C are further investigated as the title and abstract do not provide sufficient information to make an informed decision (see Table 2.5). To further investigate the articles in Category C, adaptive reading depth is used [12].

<table>
<thead>
<tr>
<th>Category</th>
<th>Action</th>
<th>Corresponding strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &amp; B</td>
<td>Consider articles potential primary studies.</td>
<td>D2 see Table 2.3.</td>
</tr>
<tr>
<td>F</td>
<td>Discard as irrelevant articles.</td>
<td>D7 see Table 2.3.</td>
</tr>
<tr>
<td>C</td>
<td>Articles will be reviewed further using adaptive reading depth.</td>
<td></td>
</tr>
<tr>
<td>D &amp; E</td>
<td>Reviewers discuss the articles and by consensus place them in one of the above categories.</td>
<td>R3 see Table 2.2.</td>
</tr>
</tbody>
</table>

Due to the quality of the abstracts and for following an inclusive strategy when dealing with the conflicts in selection, we had a large number of articles which required further review, i.e. articles in category C. This issue was solved with a trade-off by adopting a process of adaptive reading depth [12] instead of full text reading. We again performed a pilot of this procedure on a subset of five articles to develop a shared interpretation. This process is shown in Figure 2.4.

2.6.3 Experiences and reflections

A reflection on each step of our inclusion/exclusion strategy is presented. *Objective criteria assessment (post-inclusion/exclusion)*: As shown in Table 2.6 there was a large number of studies classified as irrelevant by both authors, thus solely looking at the inter-rater statistics may be misleading, e.g. in the case of this pilot study, for 1374 articles reviewers were in complete agreement to exclude the article as irrelevant. Where as for 94 articles there was some contradiction between the classification decision by the two reviewers. Out of these 94, only 17 articles were serious conflicts where one reviewer marked an article as relevant and the other as irrelevant.

Automatic search in digital databases often generates a lot of irrelevant results because of the limited search capabilities of various databases and lack of agreed upon terms to refer to concepts among other issues. In our observation the agreement on
exclusion is usually considerably higher because of these evidently irrelevant results, this in turn skews the statistics computed on inter-rater agreement and it is prone to give a false confidence in the goodness of criteria.

Results of selection in the pilot study: Out of the 1649 articles on which selection criteria were applied, 71 articles ended up in categories D and E. These articles were discussed and recategorised in one of the other categories A, F or C. The final selection results after discussion are shown in Table 2.7 where the number in parenthesis is the number of articles placed in the current category after discussion from either category D or E.

Two reviewers independently applied adaptive reading depth on 158 articles from category C. Again, inter-rater agreement was computed on the resulting classification. 70 of these articles were found relevant and 104 were rejected as irrelevant studies. So
Table 2.6: Results of applying primary criteria for study selection.

<table>
<thead>
<tr>
<th>Category ID</th>
<th>Number of articles</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>69</td>
<td>Accepted for full text reading.</td>
</tr>
<tr>
<td>B</td>
<td>23</td>
<td>Accepted for full text reading.</td>
</tr>
<tr>
<td>C</td>
<td>112</td>
<td>Need further review.</td>
</tr>
<tr>
<td>D</td>
<td>17</td>
<td>Discuss the disagreement.</td>
</tr>
<tr>
<td>E</td>
<td>54</td>
<td>Discuss the disagreement.</td>
</tr>
<tr>
<td>F</td>
<td>1374</td>
<td>Excluded from further steps in the SLR.</td>
</tr>
</tbody>
</table>

Table 2.7: Selection results for the pilot study.

<table>
<thead>
<tr>
<th>Category</th>
<th>Initial number of articles</th>
<th>Number of articles after discussion</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &amp; B</td>
<td>92</td>
<td>98</td>
<td>Accepted for full text reading</td>
</tr>
<tr>
<td>C</td>
<td>112</td>
<td>158</td>
<td>Need further review</td>
</tr>
<tr>
<td>D &amp; E</td>
<td>71</td>
<td>0</td>
<td>Recategorised after discussion</td>
</tr>
<tr>
<td>F</td>
<td>1374</td>
<td>1393</td>
<td>Excluded from further steps in SLR</td>
</tr>
</tbody>
</table>

in total we found 168 potential primary studies in the selection step of this pilot study.

Decision Rules: Having completed the systematic literature review in the pilot study, we can make some interesting comparisons regarding the choice of the strategies and loss of potentially relevant articles as shown in Table 2.8. The first column of the table shows the strategies identified in the literature review, and maps those to the strategies shown in Table 3.3 in the fourth column. The values in the second column of this table are the numbers of articles that this decision would have short-listed. The third column shows the number of primary studies eventually identified from the decision.

The comprehensiveness of the strategy employed in this pilot review, because of its inclusiveness, is evident from these results, i.e. using D7 more relevant articles were identified. Strategies D4, D5 and D6 also compete in coverage with the strategy D7 used in this study. However, following D2 and D3 which has been used in existing literature reviews would have resulted in a significant loss of relevant articles.

Both D4 and D7 are not very explicit on how to handle all the possible combinations of resulting variation in selection results. On the other hand, unique to this pilot study was categorization of articles in categories demanding different levels of attention from the reviewers based on the level of disagreement. Thus, it was rational and
fairly safe to assume that articles where one reviewer considered the article relevant and the other was not sure (as in the case of category B articles) to be included as relevant. Similarly, the articles in category D had maximum contradiction in selection and thus needed more attention and were therefore discussed. Lastly, for articles in category C no discussion would have been fruitful as both reviewers agreed that the available information (in title and abstract) is insufficient to make a decision.

Table 2.8: Impact of strategies on potential primary studies.

<table>
<thead>
<tr>
<th>Strategy ID</th>
<th>Number of articles</th>
<th>Primary studies</th>
<th>Pilot category</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Not applicable here only.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>115</td>
<td>47</td>
<td>A+B+D</td>
</tr>
<tr>
<td>D3</td>
<td>75</td>
<td>37</td>
<td>A</td>
</tr>
<tr>
<td>D4</td>
<td>264</td>
<td>65</td>
<td>A+B+C+E</td>
</tr>
<tr>
<td>D5</td>
<td>227</td>
<td>62</td>
<td>A+B+C+D</td>
</tr>
<tr>
<td>D6</td>
<td>210</td>
<td>59</td>
<td>A+B+C</td>
</tr>
<tr>
<td>D7</td>
<td>281</td>
<td>68</td>
<td>A+B+C+D+E</td>
</tr>
</tbody>
</table>

2.7 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 [17]. 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, 17] 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 2.3). 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 in post-hoc test with care: Another lesson learned from the pilot review is that the inter-rater statistics should be used carefully. As a large number of obviously unrelated articles found in automatic searches can skew the raters agreement levels and paint a more positive picture about the objectivity of the criteria. Perhaps one work around could be to apply this after the straightforward unrelated articles have been rejected. Since, the real strength of objectivity of the criteria is tested on the borderline cases and not the extreme cases of relevance.

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: Less inclusive strategies seem to lead to a loss of articles: In our analysis, we provided data showing that only if we run a highly inclusive strategy (i.e. consider categories A+B+C+D+E) the highest number of relevant articles are identified. All other strategies (e.g. only use A+B+C) would lead to a loss of relevant articles. For example, for the strategy A+B+C 9 relevant articles would not have been captured. Given that the goal is to select articles as close to the study population as possible, the most inclusive strategy should be followed. From an effort perspective, the adaptive reading took a maximum of 15 minutes per article, hence the additional effort between strategy A+B+C to strategy A+B+C+D+E is 18 hours. Considering reading the full-article (let’s assume this would take two hours) a total of 142 hours would be consumed. This further underlines the cost-effectiveness of adaptive reading depth mentioned in Lesson 4.

However, the loss of articles for strategies D4, 5 and 6 is not very significant.
2.8 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 analysis, we believe that the proposed review process is a good candidate solution to become widely adopted to make study selection in systematic literature studies 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 future work one should investigate whether the loss of articles in less inclusive strategies would have led to a change of the conclusions.
REFERENCES

2.9 References


Chapter 3

Software process simulation: A systematic review on rigor, relevance, and impact

3.1 Abstract

Software development processes are a set of complex interactive activities that are difficult to analyse with static analytical process models. Software process simulation on the other hand captures the dynamic behavior for the evaluation of “what-if” scenarios for different purposes, such as predictions and evaluating process improvement alternatives. This study aims at identifying, assessing and aggregating empirical evidence on usefulness of software process simulation in real-world software (industrial and open source) development. A systematic mapping study and review of published literature were conducted. System dynamics and discrete event simulation are the two most commonly used approaches in real-world software development. Simulation has been applied for purposes (planning, operational control etc.) and scopes (from a phase of lifecycle to long term evolution etc.). The primary studies scored poorly on rigor and relevance criteria. Very few performed some initial evaluation of the simulation model for the intended purposes. There is lack of evaluation of usefulness of software process simulation in relation to simulation purposes. The cost of applying it in the real-world is also not reported. Furthermore, the need for improvements in reporting simulation studies is highlighted.
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

### 3.2 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 deliver quality products. Both academia and industry are striving to find ways for continuous software process improvement (SPI). There are numerous SPI frameworks and methodologies available today [21, 19, 20] but they all have one common challenge: the cost of experimenting with the process change. It is widely believed that software process simulation modelling (SPSM) can help in predicting the benefits and repercussions of a process change [10]. Thus, enabling organizations to make more informed decisions and reduce the likelihood of failed SPI initiatives. Simulations have been widely used in various domains and there is considerable literature that focuses on SPSM.

There are a few secondary studies on the subject that have scoped the research available on the topic [14, 32, 27, 28, 29, 30, 1]. The systematic reviews by Zheng et al. [27, 28, 29, 30] and de França and Travassos [1] included simulation studies without a specific focus on industrial application. In essence, this study aims to aggregate empirical evidence on evaluation of these simulation methods in real-world settings based on a systematic assessment of rigor and relevance [34]. With real-world settings we mean industry and open source development. Rigor refers to the completeness of the reporting to be able to judge the scientific rigor of the article. Relevance refers to the practical impact of the study in terms of usefulness. The usefulness is seen in connection to the purpose of simulation. This will help in identifying suitable simulation approaches for various proposed purposes along with their applicable context.

The research methodologies used in this study are systematic mapping [35] and systematic review [24].

The remainder of the chapter is structured as follows: Section 3.3 presents the related work. Section 3.4 explains our research methodology. Section 3.5 shows the results of the mapping, followed by the review results in Section 3.6. Section 3.7 discusses the results of the mapping and review study. Section 3.8 concludes the paper.

### 3.3 Related work

Using the search strings and the process reported in Section 3.4.2 we identified three existing and recent reviews of the SPSM literature. One SLR primarily focuses on software risk management by Liu et al. [32] (published in one article) and the other one is a two phased SLR which is reported in four articles by Zhang et al [27, 28, 29, 30]. Further papers were identified by forward snowballing applied to the systematic
reviews identified in the search, namely de Francça and Travassos [1] and Zhang et. al. [31].

From the criteria sets identified by Kitchenham and Charters [24] to assess an existing review, we used the detailed checklist proposed by Khan et al. [12] and the general questions recommended by Slawson [11]. The aim is to evaluate the reviews on their objectives, coverage (data sources used, restrictions etc.), methodology, data extraction, quality assessment, analysis, validity and reporting. Our evaluation results of using these criteria on the existing literature reviews are presented below:

**What are the review’s objectives? Is there an important SE question addressed in the review?** Zhang et. al [27, 30] have used six broad questions to scope the field of SPSM. Similarly, Liu et. al [32] also seek answer for five broad scoping questions but focusing on SPSM’s use in Software Risk Management. de Francça and Travassos [1] characterized the models in terms of model type, structure, verification and validation, and how the results of the simulation were presented in terms of visualization.

Zhang et. al [27, 30, 32] have similarities to a systematic mapping study [35]. Zhang et al. [30] also remark in the study’s conclusion that their systematic literature review “is also a kind of mapping study”. Therefore, in this study we utilize the studies from real-world settings to assess and aggregate empirical evidence to answer a more focused question i.e. to establish usefulness of SPSM. The study by de Francça and Travassos [1] is different in that it considers verification and validation of models, and hence has an element of judging the quality of the simulation models being investigated. Another difference is the consideration of all simulation purposes (e.g. architecture).

**What sources were selected to identify primary studies? Were there any restrictions?** In the initial phase, Zhang et. al [27] performed a manual search of renowned venues for SPSM literature. In the second phase, it was complemented with an electronic search in IEEE, Science Direct, Springer Link and ACM, covering literature from 1998-2007 [30]. Actual search strings used in different databases are not available and it is not mentioned whether the search was performed in full-text or not.

de França and Travassos [1] used Scopus, EI Compendex, Web of Science and developed their search string by defining population, intervention, comparison, and outcome. Their search was combining different terms used for simulation, such as agent based simulation, process simulation, and so forth. However, it is important to note that they also included architecture simulation, hence their review focused on simulation in software engineering as a whole.

In all three studies, we believe that potentially relevant sources for the Management and Business literature were not included in the search. Zhang et al. [30] 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 Software Engineering 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. Furthermore, since the time when these reviews were conducted, more research has been published on the subject and given the recent trend of empirical research we had a high likelihood of finding more studies from real-world settings.

**Are the inclusion and exclusion criteria in the review described and appropriate and how were they applied?** Zhang et al. studied [27, 30] software process simulation in a two step process: initial and final selection of papers. It is broadly defined what will be considered relevant for inclusion. The steps and information from articles used (e.g. title or abstract etc.) to make a decision is also explicitly stated.

Liu et al. [32] assessed the relevance of articles based on titles, abstracts and keywords. They reported on cases where a articles were rejected based on full-text reading.

de França and Travassos [1] included all simulation studies that were related to the software engineering domain. In their selection of studies one reviewer was conducting the selection first, and the remaining two reviewers cross-checked the selection.

There is some level of detail missing in both articles but that may be because of the page limitations in the conference proceedings. A potential threat to the validity of the studies by Zhang et al. and Liu et al. is that in both studies only one reviewer out of three applied the inclusion and exclusion criteria to find the primary studies. The results of selection are therefore highly dependent on one reviewer’s subjective judgement and may be biased in this regard. With such a large amount of literature that reviewer had to go through for these two broad studies, a reviewer is likely to make mistakes or overlook important information. If only one reviewer is doing the selection there is no safety net and any mistake can result in missing out a relevant article [25]. In the article by de França and Travassos [1] several reviewers were involved to increase the validity of study selection.
We have decided to include two reviewers and other preventive measures (discussed in detail in Section 3.4.4) to minimize the threats of excluding a relevant article.

**What quality criteria were used to assess the quality/validity of included studies?**

Zhang et al. [27, 30], use the same criteria for evaluation of the studies that comprises of four basic questions and then questions particular to each of the four classes of research articles are stated. Liu et al. [32] used a subset of the criteria from [27] and adapted it for their study.

In the Zhang et al. studies [27, 30] and Liu et al. [32], it appears that the quality assessment results were not used in selection of studies. It is also unclear if the results of the quality assessment influenced the analysis and conclusions drawn in the study. Since it is a mapping study, a qualitative sensitivity analysis could have been performed, for example, what simulation approaches have been applied for a certain purpose in high quality studies and how are the results different from overall study results.

de Franc¸a and Travassos [1] reported that no quality appraisal was done in their studies. However, 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.

In both studies only one reviewer assessed the quality of studies. Although, Zhang et al. report that a second reviewer applied the quality assessment criteria on a subset of the studies. However, the results on inter-rater agreement are not reported so it raises some validity concerns. In de Franc¸a and Travassos no quality check is reported.

**How were the data extracted from the primary studies?** In [27], a detailed data extraction form is provided however it is not clear if more than one reviewer extracted data from the studies. The same data extraction form is used in the second phase of their study [30], however in this phase of the literature review a second reviewer also extracts data from 15 percent of the studies for validation. However, neither the level of agreement between the two reviewers nor the mechanism used to resolve conflicts is presented. Furthermore, it is not clear why a pilot was not performed before the actual extraction.

In [32], a data extraction form is provided and it appears that only one reviewer was responsible for the extraction.

de Franc¸a and Travassos [1] reported on their data extraction form, but did not report on the procedure followed in data extraction.

**How were the data synthesised?** Both Zhang et al. [27, 30] and Liu et al. [32]
used classifications and frequency of occurrence in extracted data to identify trends in the research. No details of which qualitative analysis methods were used are reported in the articles. de França and Travassos [1] used tabulation to illustrate which domains simulation has been used for; illustrate the frequency of verification and validation approaches used; presenting the frequencies of how simulation results were visualized.

Were the basic data/studies adequately described? Were the differences between the studies investigated? The list of primary studies for the second phase of Zhang et al. review [27, 28, 29, 30] has not been reported in the articles, which is likely because of the page limitation in conferences (the web-page they have provided does not work http://systematicreviews.org). Liu et al. [32] on the other hand included a list of primary studies in the article. In de França and Travassos [1] no list of primary studies was reported.

How were differences between studies investigated? How were the data combined? Was it reasonable to combine the studies? The primary approach used to aggregate results was frequency analysis in terms of how many studies investigated a specific aspect of simulation.

Do the conclusions flow from the evidence? Yes, the mapping results represent the studied purposes, modelled scopes and the tools used in the primary studies.

How sensitive are the results to the way the review has been done? The main threat to validity of these studies is the use of a single reviewer for inclusion and exclusion of studies, quality assessment and data extraction. The mechanism of how the disagreements were resolved in the phases where more than one reviewer were involved is not documented. This along with the limitations identified in answers to the above questions make the results of these reviews highly susceptible to bias. Zhang et al. [27, 28] state as a limitation of their study is that the process they used to conduct the review is the one recommended for Ph.D. candidates. They acknowledge that “The main limitation of our review is that the process recommended for PhD candidates is not as rigorous as that adopted by multiple-researchers.” [27]. Similarly, the actions taken to reduce this limitation “a quality assurance check by the secondary researcher and review of the research protocol, and the results by an external researcher.” are insufficient to increase credibility of the review. Because an external researcher’s review of the protocol cannot ensure a consistent understanding and application throughout the review. Likewise if the results are in-line with the external researcher’s expectations it is very likely that such a ‘review’ by an external expert is not of much value (unless of course if the external expert followed the chain of evidence and replicated how it was
identified and aggregated).

Similarly, in [32], the entire review was basically 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." where as the other two reviewers had a more supervisory role which although is better than doing a review alone but does not eliminate serious validity concerns for the review results.

de França and Travassos [1] used several reviewers in the selection phase, but did not report on how it made the data extraction and classification of studies more reliable.

Similarly looking at the citations to these reviews, we found one more article by Zhang et. al. [31] that is based on the results of their systematic reviews and claims to have done some impact analysis of SPSM research. However, the following conclusions in the article are not linked to 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.
- "the impact of SPS research is very difficult to quantify. 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”.

Given the limitations identified in Zhang et al. review, it is not very assuring when they claim “the evidence of this research is solidly based on a Systematic Literature Review (SLR) and its extension. In 2008 and 2010, we reported a two-stage SLR on software process simulation [50, 52], which aimed to assess how SPS has evolved over the decade (1998-2007)” [31].

Zhang et al. [31] present an overview of software process simulation and a historical account/timeline of SPSM research capturing who did what and when. The “case study” reported in this article to supplement the “impact” analysis is at best anecdotal. Lastly, 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”.

They [31] conclude that “the collaboration between researchers and practitioners is important (if not indispensable) to produce a high quality process simulation model that is able to realistically capture the problem” suggesting that the simulation community has been unable to transfer technology to practice. At the same time they report
that the SPSM research has had “a significant impact on practice”. To determine which one of these statements holds traceable evidence is needed, which we provided by assessing rigor and relevance of studies in a systematic manner, and by analysing how the evidence in the articles connects to the purpose of using the simulation models.

Kellner et al. [14] provide an overview of the SPSM literature. Their study was published in 1999 and there is considerable new literature available on the topic. We use their work to explore how the research from real-world application of SPSM has used the simulation approaches for the purposes identified in their study.

In this literature review we attempt to address the limitations of the existing literature reviews by following a more explicit strategy during the various phases of the review. Contrary to the existing reviews we have focused on assessing and aggregating the empirical evidence and scoping the real-world SPSM research, which was not the goal of the earlier reviews. The contributions of this study are as follows:

- Use of an explicit and systematic process to identify and evaluate the real-world research on SPSM.
- Using pilots, inter-rater agreement statistics and a documented process to resolve differences this study reduces the bias in various steps of selection, extraction and evaluation of the primary studies.
- No time or venue limitations on the published literature and the inclusion of several databases that cover computer science, software engineering and management literature ensured broad coverage.
- We evaluate the rigor of the primary studies and their relevance to real-world settings. This was used to assess the evidence reported in the primary studies to address which simulation approach is useful for what purpose and in which context.
- We identify the limitations of current research in a systematic and traceable manner and present guidelines on what needs to be done in future research to improve the impact of SPSM.
- The systematic mapping study also only focuses on real-world research and scopes the applied research in SPSM.

### 3.4 Research methodology

This study aims to explore the state of knowledge regarding use of software process simulation in real-world this was achieved by conducting a software mapping study using the guidelines from Petersen et al. [35].
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.

3.4.1 Research questions

The following sections present the questions related to the systematic mapping study and the systematic literature review.

Mapping Questions

To answer which approaches to simulation have been used in real-world context and what was the scope and purpose of modelling we answered the following question:

• MQ 1: For which purposes has SPSM been used in a real-world setting?
  
  – MQ 1.1: What was the scope of the simulation model in these studies?
  
  – MQ 1.2: Which approaches of simulation modelling were used in these studies?

To identify the thought leaders in applied SPSM research and to develop insights into what is currently considered valuable and also to provide what is the future directions of research we posed the following questions:

• MQ 2: Which are the most influential real-world articles on simulation?

• MQ 3: Who are the most influential researchers on applying simulation in the real-world?

Review Questions

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

• RQ 1: What evidence has been reported that the simulation models achieve their purposes in the real-world settings?
3.4.2 Need for 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 Almeida et. al [26]. The search was conducted in the databases identified in Table 3.1 using the following search string with two blocks joined with a Boolean ‘AND’ operator:

\[(\text{software AND process AND simulation}) \text{AND} (\text{“systematic review” OR “research review” OR “research synthesis” OR “research integration” OR “systematic overview” OR “systematic research synthesis” OR “integrative research review” OR “integrative review” OR “systematic literature review”})\]

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 ([27, 28, 29, 30]) and ([32]). We also read the titles of articles that cite these reviews and found two more relevant article [31, 1]. In Section 3.3 we have already discussed in detail the limitations of these six articles. We have also discussed the novel contributions in our study and how we have attempted to overcome the shortcomings in these reviews.

3.4.3 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 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.

The search in these databases was restricted to title, abstract and keywords except in Google Scholar where it was only done in the title of the publication. This restriction was made for practical reasons as Google Scholar is more of a search engine than a bibliographic database therefore we made a trade-off in getting a broader coverage by
Table 3.1: Digital databases used in the study.

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

using it but restricting to search in titles only to keep the number of hits practical for the scope of this study.

Keywords

Starting with the research questions suitable keywords were identified using synonyms, encyclopaedia of SE [33] and seminal articles in the area of simulation [14]. Following are the keywords used to formulate the search strings:

- **Population**: Software process or phase thereof. *Alternative keywords*: Software project, software development process, 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.
- **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. Following is the resulting search string:

\[(\text{software} \text{'Proximity Op' process}) \text{OR (software} \text{'Proximity Op' project})\text{AND (simulat\star OR “dynamic model” OR “system dynamic” OR “state based” OR “rule based”}} \]
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

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.

Figure 3.1 provides an overview of search results and selection procedure (discussed in detail in Section 3.4.4) applied in this review to identify and select primary studies.

3.4.4 Study selection criteria and procedure

Before the evaluation of relevance all search results were subjected to following 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 later on when the full-text of the articles were read handled).

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 lot of duplicates. Also, in Google Scholar we had no mechanism to keep out grey literature from 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, preliminary criteria were used to remove the obviously irrelevant articles.

Preliminary criteria

Preliminary criteria were applied primarily 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.
Excluding any articles related to simulation of hardware platforms.

2. Exclude any articles related to use of simulation software, e.g. simulating manufacturing or chemical process or transportation, etc.

3. Exclude any articles related to use of simulation for evaluation of software or hardware reliability and performance etc.

4. 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...
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

the SPSM used for training and education in industry from those in a pure 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 the principal reviewer alone. 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 judgement 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 simulation of 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 domain were included in this review.

Before performing the actual inclusion and exclusion of studies, a pilot selection was performed. This step was conducted by two reviewers. First, both researchers read the criteria and using “think aloud” protocol applied it on 3 randomly selected articles. This helped to develop a common understanding of the criteria. Later on, both researchers applied the same criteria on 20 randomly selected articles independently. It was decided that articles will be labelled as: Relevant, Irrelevant or Uncertain (if available information i.e. title and abstract, is inconclusive). The results of this pilot are shown in Table 3.2.

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. Both reviewers independently applied inclusion exclusion criteria on the 1649 articles (that had passed the preliminary criteria used for initial screening).

Inclusion and exclusion criteria were applied independently by the primary and secondary reviewers using three choices to categorize each article: relevant, irrelevant or uncertain as done previously in the pilot selection. When the results of applying
Table 3.2: Results of pilot selection.

<table>
<thead>
<tr>
<th></th>
<th>Relevant</th>
<th>Uncertain</th>
<th>Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reviewer 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevant</td>
<td>6</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Uncertain</td>
<td>-</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Irrelevant</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
</tbody>
</table>

Inclusion and exclusion criteria were acquired from both reviewers we had these possibilities (as shown in Table 3.3) of agreement or disagreement between the reviewers.

In Table 3.3 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:

- 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 was certain about the relevance and the other was inconclusive) we considered it appropriate to include such studies for full-text reading.
- On the other hand, articles in category F were 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 supplement 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 considered an article relevant and other considered 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
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

Table 3.3: Different possible scenarios from study selection.

<table>
<thead>
<tr>
<th>Reviewer 1</th>
<th>Relevant</th>
<th>Uncertain</th>
<th>Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

The results of this phase are summarized in Table 3.4 where the third column shows the final total of articles once the articles in category D and E were discussed and reclassified.

Table 3.4: Inclusion and exclusion criteria results.

<table>
<thead>
<tr>
<th>Category ID</th>
<th>Number of articles</th>
<th>Total number of articles post discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>69</td>
<td>75</td>
</tr>
<tr>
<td>B</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>C</td>
<td>112</td>
<td>158</td>
</tr>
<tr>
<td>D</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>1374</td>
<td>1393</td>
</tr>
</tbody>
</table>

Table 3.5 reports shows high agreement on outcome of applying basic criteria on articles. This shows a shared understanding and consistent application of the criteria on the articles.

*Adaptive reading for articles in category C:*

Based on the titles and abstracts of articles, we often lacked enough information to make a judgement about the context and method of the study. Therefore we had 158 articles that required more information for decision making. Many of the existing literature reviews exclude such articles where both reviewers do not consider a study.
relevant [22]. However, as we decided to be more inclusive to reduce the threat of excluding relevant research we decided to further investigate such studies.

As the number of articles in this category was quite large (158 articles) and we already had a sizeable population of potential primary studies (75 and 23 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 for the purpose of this study. The inter-rater agreement was fairly high for this step as presented in Table 3.6. All articles with conflicting decision between the two reviewers were taken to the next step for full-text reading.

Table 3.5: Cohen’s Kappa and percent agreement on application of basic selection criteria.

<table>
<thead>
<tr>
<th>Percent Agreement</th>
<th>Cohen’s Kappa statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>94.11</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Table 3.6: Cohen’s Kappa and percent agreement on application of Adaptive reading.

<table>
<thead>
<tr>
<th>Percent Agreement</th>
<th>Cohen’s Kappa statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.84</td>
<td>0.53</td>
</tr>
</tbody>
</table>

**Advanced criteria**

This is related to actual data extraction, where 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 3.4.4 and Section 3.4.4) but this time reading the full-text of the articles. Also exclude the conference articles that have been subsequently extended to journal articles (that are more likely to have more details).

For practical reasons these 168 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. The further Section 3.4.7 presents the data extraction form used in this study, the results of the pilot and the actual data extraction performed in this study.

**3.4.5 Study quality assessment criteria**

The criteria used in this study was adapted from Ivarsson and Gorschek [34] to fit the area of SPSM. We dropped ‘research methodology’ and ‘context’ as criteria from the relevance category as we only included real-world studies in this review therefore these fields were redundant. Furthermore, experiment and case study are often not used in the same sense in the SPSM context that would have made it very difficult to classify the research methods employed in the primary studies.

**Scoring for rigor**

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

- **Description of context**
  1. If the description of the context covers all of the four context facets [36] then the score is ‘1’.
  2. If the description of the context covers at least two of the context facets then the score is ‘0.5’.

72
3. If less than two facets are presented then the score is ‘0’.

- **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 defined 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 [18] (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’).
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

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 3.7. Generally we had 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 validity of the results as all the conflicts were resolved by discussion and referring back to the full-text of the publication. The major reason for this low value was that one reviewer marked several articles for discussion where it was not very obvious whether the users actually used the simulation model for the intended purpose, while the other reviewer made a decision based on the available information in the article. This has been reflected in quality assessment criteria for repeatability.

Table 3.7: Cohen’s Kappa and percent agreement on quality assessment of primary studies.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Percent Agreement</th>
<th>Cohen’s Kappa statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context description</td>
<td>80.88</td>
<td>0.64</td>
</tr>
<tr>
<td>Study design description</td>
<td>77.94</td>
<td>0.29</td>
</tr>
<tr>
<td>Validity threats discussion</td>
<td>94.12</td>
<td>0.56</td>
</tr>
<tr>
<td>Subjects/Users</td>
<td>89.70</td>
<td>0.19</td>
</tr>
<tr>
<td>Scale</td>
<td>67.65</td>
<td>0.34</td>
</tr>
<tr>
<td>Model Validity</td>
<td>88.23</td>
<td>0.81</td>
</tr>
</tbody>
</table>

The results of quality assessment of primary studies after consensus are presented Table 3.8.

Table 3.8: Rigor, relevance and model validity scores for the primary studies

<table>
<thead>
<tr>
<th>Reference</th>
<th>Context description</th>
<th>Study design description</th>
<th>Validity discussion</th>
<th>Subjects</th>
<th>Scale</th>
<th>Model validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[53]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[41]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Continued on next page
Table 3.8 – Continued from previous page

<table>
<thead>
<tr>
<th>Reference</th>
<th>Context description</th>
<th>Study design description</th>
<th>Validity discussion</th>
<th>Subjects</th>
<th>Scale</th>
<th>Model validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[40]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[44]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[45]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[72]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[84]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[78]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[39]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[89]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[59]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[48]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[97]</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[99]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[63]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[94]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[79]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[101]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[75]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[52]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[74]</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[66]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[65]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[49]</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[70]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[61]</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[36]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[13]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[77]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>[42]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[81]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>[80]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[88]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>[47]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>[43]</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Continued on next page
Table 3.8 – Continued from previous page

<table>
<thead>
<tr>
<th>Reference</th>
<th>Context description</th>
<th>Study design description</th>
<th>Validity discussion</th>
<th>Subjects</th>
<th>Scale</th>
<th>Model validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[60]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>[93]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[50]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[86]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[90]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[67]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>[82]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[71]</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[64]</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[76]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[102]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[98]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[103]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[56]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[87]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[105]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[91]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[46]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[104]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[100]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[57]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[55]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[85]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[69]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[73]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[54]</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[62]</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[92]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>[45]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[96]</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[58]</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>[51]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[106]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>
3.4.6 Scoring model validity

To assess the credibility of models 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, or did sensitivity analysis; b) Checked the model against reference behaviour or compared model output with past data 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. Co-hen’s Kappa value for inter-rater agreement for “Model Validity” is 0.81 (see row 7 in Table 3.7). This shows 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.

3.4.7 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 3.4.4.
- **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.
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

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 [14]. 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 analysed and mapped to the scopes identified by Kellner et al. [14]. This mapping was also reviewed by second reviewer for all the primary studies.

### 3.4.8 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 use of a generic search engine, Google Scholar and not restricting the search results by time of publication or venue in any database.

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 was guided by an explicit simple 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 believe also minimized the threat of excluding a relevant study. Selection of articles based on basic criteria was
done by both reviewers and an explicit strategy based on Petersen and Ali [22] was employed.

All the selection of studies, data extraction procedures and quality assessment criteria were piloted and the results are presented in the paper. 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 reviewers bias and dependence of review results on personal judgements. This also increased the repeatability of the review.

Inter-rater agreement was also calculated for such activities and is discussed in the paper where both reviewers performed a task e.g. application of basic criteria for selection and quality assessment. All the conflicts were resolved by discussion and reviewing the primary studies together. So even in the cases e.g. when the value of agreement is low in quality assessment we minimized this threat by reviewing all the conflicting scores and also elaborated why the disagreements existed.

Studies are in some cases based on Ph.D. theses, e.g. [43, 44]. 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 numbers could change, however, the principle conclusion would not be different.

Data analysis that resulted in classification of purpose and scope for the models also involved both reviewers. This way we reduced the threat of classifying the data incorrectly.

### 3.5 Mapping results

#### 3.5.1 Purpose

Table 3.9 provides an overview of real-world simulation studies relating to the purposes defined in [14]. The clear majority of studies used simulation for planning purposes (34 studies), followed by process improvement (20 studies), and training and learning (18 studies). In comparison, only a few studies used simulation for control and operational management.

*Planning:* In planning simulation has been used for decision support [40, 97, 66,
Table 3.9: Purpose of simulation in the primary studies.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>No. of articles</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control and Operational Management Planning</td>
<td>7</td>
<td>[75] [65] [49] [105] [104] [85] [45]</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>[41] [40] [45] [72] [78] [39] [59] [48] [97] [99] [79] [101] [75] [66] [70] [61] [36] [13] [62] [42] [81] [80] [43] [60] [86] [64] [76] [103] [46] [85] [73] [92] [96] [51]</td>
</tr>
<tr>
<td>Process Improvement and Technology Adoption</td>
<td>20</td>
<td>[53] [84] [89] [63] [74] [77] [88] [60] [93] [50] [90] [82] [71] [102] [98] [56] [87] [100] [69] [58]</td>
</tr>
<tr>
<td>Training and Learning</td>
<td>18</td>
<td>[44] [94] [52] [65] [13] [62] [47] [60] [93] [67] [91] [57] [55] [54] [62] [96] [51] [106]</td>
</tr>
</tbody>
</table>

61, 59, 92, 42, 43, 36, 76] (e.g. in relation to staffing and allocation of effort [36, 42, 61], connecting decisions and simulating them in relation to business outcomes [59], and requirements selection [92]). Furthermore, simulation has been used by many studies for estimation [48, 13, 41, 75, 51, 81, 78, 86, 72, 79, 70, 99, 80, 51, 62, 103, 96, 46], some examples are cost and effort estimation [75, 51, 79], schedule estimation [41], release planning, and fault/quality estimation [99, 72, 39].

Process Improvement and Technology Adoption: Studies in this category used simulation to evaluate alternative process designs for process improvements [74, 87, 98, 84, 89, 102, 63, 71, 69]. As an example, [98] 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. [102, 63, 71]), as well as investigating specific technology adoptions to the process [88, 58, 90, 77, 100, 50, 56]. Examples of technology adoptions were introduction of test driven development (TDD) [77, 100], comparison of manual vs. model-based techniques [50], or use of different quality assurance approaches or architectural styles [56].

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. [94, 62, 60, 13, 106, 96, 62, 67, 44, 93, 54, 51]). Examples are to understand the effect of requirements overload on bottlenecks in the whole process [62] or understanding open source system development [67], assess what factors make global software development successful [96], or understanding the effects of creeping requirements [13].

Control and Operational Management: Only few studies used simulation for control and operational management [65, 85, 49, 104, 45, 75, 105]. As an example, several studies assessed whether a project is likely to meet its expected deadline [49, 104], or whether an individual iteration is able to meet the deadline [45]. Studies [75, 105] used simulation for progress status assessment.

3.5.2 Scope

Table 3.10 defines categories for scope that a simulation study can have.

The clear majority of studies focused on individual projects (46 studies). In comparison, few studies looked at the portion of a lifecycle (13 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 (2 studies).

Projects: Studies investigating projects and their development lifecycle looked at different lifecycles, e.g. related to CMMI processes [65], and extreme programming (XP) projects [77]. One study explicitly stated that they are focusing on new software development [36], while others in connection with the software lifecycle (excluding requirements and maintenance phase.), see e.g. [41, 44]). The remaining studies only specified that they are looking at the overall development lifecycle of projects without further specification of type and scope/boundaries.

A Portion of the Lifecycle: Studies looking at a portion of a lifecycle investigated focused on the maintenance process solely [53, 87, 91], software reliability life cycle [99], requirements and test [47], quality assurance processes [50, 90], requirements [57], release processes [46, 62], and processes for components off the shelf (COTS) selection and use [93].

Long term product evolution: Researchers have looked at the general long-term evolution of products without more specific classification [102, 54, 106] while [92] looked at market-driven requirements processes from a long-term perspective.

Long term organization: From an organizational perspective, studies investigated factors influencing process leadership [84], processes at an organizational level [59], and CMMI process in relation to organizational business concerns [52].

Concurrent Projects: Two parallel projects have been simulated by [40], while multiple concurrent projects have been simulated by [98].
Table 3.10: Scope of simulation in the primary studies.

<table>
<thead>
<tr>
<th>Scope</th>
<th>No. of articles</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>A portion of life-cycle</td>
<td>13</td>
<td>[53], [99], [63], [49], [47], [93], [50], [90], [87], [91], [46], [57], [62]</td>
</tr>
<tr>
<td>Development project</td>
<td>46</td>
<td>[41], [44], [45], [72], [78], [39], [89], [48], [97], [94], [79], [101], [75], [74], [66], [65], [70], [61], [36], [13], [77], [42], [81], [80], [88], [43], [60], [86], [67], [82], [71], [64], [76], [103], [56], [105], [104], [100], [55], [85], [69], [73], [45], [96], [58], [51]</td>
</tr>
<tr>
<td>Concurrent projects</td>
<td>2</td>
<td>[40], [98]</td>
</tr>
<tr>
<td>Long term evolution</td>
<td>4</td>
<td>[102], [54], [92], [106]</td>
</tr>
<tr>
<td>Long term organization</td>
<td>3</td>
<td>[84], [59], [52]</td>
</tr>
</tbody>
</table>

3.5.3 Simulation Approaches

Table 3.11 shows the simulation approaches used by the identified studies. System dynamics (SD) is the most commonly used simulation approach (26 studies), followed by discrete event simulation (DES) with 16 studies. A total of 10 studies combined different simulation approaches. Only few studies used approaches such as petri nets (PN) (6 studies) and Monte Carlo (2 studies).

Hybrid simulation models were mostly combining SD and DES [74, 13, 77, 57, 45, 96, 51]. Furthermore, qualitative simulation was combined with other models such as SD, or used to abstract from a quantitative model (cf. [106, 91]). Furthermore, models combined stochastic and deterministic aspects in a single model [60].

Others include a variety of models that used approaches from control theory [103], Copula methods [104], Markov Chains [58], agent-based [55], while others 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 [93, 86, 85]. Other simulation tools used were Powersim [47], iThink [72]. One study used a self-developed tool [73]. The majority of SD studies did not report the tool they used.

**DES:** For DES two studies used Extend [90, 46] and two studies used Process
Table 3.11: Simulation approaches in the primary studies.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Number of Articles</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>16</td>
<td>[53], [78], [99], [63], [79], [101], [52], [49], [88], [90], [87], [46], [62], [92], [89], [82]</td>
</tr>
<tr>
<td>Hybrid</td>
<td>10</td>
<td>[74], [13], [77], [60], [91], [57], [45], [96], [51], [106]</td>
</tr>
<tr>
<td>Monte Carlo</td>
<td>2</td>
<td>[61], [56]</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
<td>[103], [104], [58], [55], [59], [75], [50], [65]</td>
</tr>
<tr>
<td>PN</td>
<td>6</td>
<td>[39], [97], [66], [81], [80], [69]</td>
</tr>
<tr>
<td>SD</td>
<td>26</td>
<td>[41], [40], [44], [45], [72], [84], [48], [94], [70], [36], [42], [47], [43], [93], [86], [67], [71], [64], [76], [102], [98], [105], [100], [85], [73], [54]</td>
</tr>
</tbody>
</table>

Analysis Tradeoff Tool (PATT) [78, 79]. Further tools used were Micro Saint [53], RSQsim [92], Telelogic SDL modeling tool [62] and SoftRel [99]. For two studies simulation models were programmed from scratch in Java [52] and C++ [49]. One study used Statemate Magnum [89] and Nakatani [82] did not specify the tool used.

**Hybrid:** For hybrid simulation a variety of tools have been used, such as DEVSim++ (APMS) [13], Extend [57], iThink [60], Little-JIT [51], QSIM [91], and Venusim [106]. One study used a self-developed model written in Smalltalk [77].

**PN:** One study documented that they developed the model from scratch using C [39], others did not report on the tools used.

**Monte Carlo:** No information about tools has been provided.

### 3.5.4 Most influential real-world articles

Table 3.12 shows the ranking of the most influential articles according to their citations normalized by years of publication.

The top ranked article is “Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation” authored by Susan Ferreira, James Collofello, Dan Shunk, and Gerald Mackulak. The article is also the newest article in the set, being published in 2009.

Looking at the year of publications overall, the majority of the articles (7 out of 10) have been published around the year 2000 (plus/minus one year). The oldest article titled “The Dynamics of Software Project Staffing: A System Dynamics Based Simula-
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

tion Approach” and authored by Tarek K. Abdel-Hamid has been published in 1989, and is ranked as number 2.

All common simulation approaches are represented among the top ten articles, system dynamics being represented by 5 studies [42, 64, 86, 54, 102], discrete event simulation by 3 studies [62, 49, 89], hybrid simulation by 2 studies [60, 74]. This indicates that from a scientific impact perspective SD, DES as well as hybrid simulation have high impact studies, while SD clearly has the majority with 5 out of 10.

All purposes are represented by the top 10 articles. The majority of the most cited articles (5 articles) was focused on process improvement and technology adoption [74, 89, 64, 60, 102]. Planning is presented by three studies [42, 86, 62]. Control and operational management and training and learning are represented by one study each, cf. [49, 54] respectively.

Table 3.12: Top ten most cited articles normalized by years of publications.

<table>
<thead>
<tr>
<th>Title</th>
<th>Ref.</th>
<th>Cit.</th>
<th>Year</th>
<th>Cit. per year</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation”</td>
<td>[60]</td>
<td>36</td>
<td>2009</td>
<td>9.00</td>
</tr>
<tr>
<td>“Stochastic simulation of risk factor potential effects for software development risk management”</td>
<td>[64]</td>
<td>72</td>
<td>2001</td>
<td>6.00</td>
</tr>
<tr>
<td>“Using simulation to analyse the impact of software requirements volatility on project performance”</td>
<td>[86]</td>
<td>69</td>
<td>2000</td>
<td>5.31</td>
</tr>
<tr>
<td>“Software process simulation to achieve higher CMM levels”</td>
<td>[89]</td>
<td>72</td>
<td>1999</td>
<td>5.14</td>
</tr>
<tr>
<td>“Exploring bottlenecks in market driven requirements management processes with discrete event simulation”</td>
<td>[62]</td>
<td>61</td>
<td>2001</td>
<td>5.08</td>
</tr>
<tr>
<td>“Assessing staffing needs for a software maintenance project through queueing simulation”</td>
<td>[49]</td>
<td>44</td>
<td>2004</td>
<td>4.89</td>
</tr>
<tr>
<td>“Application of a hybrid process simulation model to a software development project”</td>
<td>[74]</td>
<td>53</td>
<td>2001</td>
<td>4.42</td>
</tr>
<tr>
<td>“Modelling A Software Evolution Process: A Long-term Case Study”</td>
<td>[54]</td>
<td>52</td>
<td>2000</td>
<td>4.00</td>
</tr>
<tr>
<td>“Software Process White Box Modelling for FEAST/I”</td>
<td>[102]</td>
<td>46</td>
<td>1999</td>
<td>3.28</td>
</tr>
</tbody>
</table>
3.5.5 **Most influential real-world researchers**

In total 81 articles are included in this research, these articles were authored by a total of 33 authors. The top 10 researchers ranked according to their impact as the sum of normalized citations for individual publications is shown in Table 3.13. The most productive researchers among the set have been Abdil-Hamid, Raffo and Pfahl, with Abdil-Hamid being ranked as number one.

Rank 1 to 4 have around twice as many normalized citations in comparison to ranks 4 to 10. Another interesting observation is that with only two publications, Collofello and Mackulak are on ranks 2 and 3.

Table 3.13: Top ten most influential authors.

<table>
<thead>
<tr>
<th>Author</th>
<th>No. of articles</th>
<th>Sum of normalized citations (Citations per year) for individual publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdel-Hamid, Tarek K.</td>
<td>7</td>
<td>17.57</td>
</tr>
<tr>
<td>James S Collofello</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Gerald T Mackulak</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Raffo, David M.</td>
<td>5</td>
<td>13.98</td>
</tr>
<tr>
<td>Pfahl, D.</td>
<td>4</td>
<td>7.65</td>
</tr>
<tr>
<td>P. Wernick</td>
<td>2</td>
<td>7.28</td>
</tr>
<tr>
<td>M. M. Lehman</td>
<td>2</td>
<td>7.28</td>
</tr>
<tr>
<td>Juan F. Ramil</td>
<td>2</td>
<td>6.73</td>
</tr>
<tr>
<td>Karl Lebsanft</td>
<td>2</td>
<td>6.62</td>
</tr>
<tr>
<td>Martin Höst</td>
<td>2</td>
<td>6.36</td>
</tr>
</tbody>
</table>

3.5.6 **Cross analysis purpose and scope**

Table 3.14 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 purpose.

A portion of the lifecycle was primarily investigated for process improvement and technology adoption (6 studies) and training and learning (5 studies). A few studies investigated planning (3 studies), and only one individual study focused on control and operational management.

Overall, research on concurrent projects is scarce. One study investigated concurrent projects for planning, and one for process improvement and technology adoption.
## Table 3.14: Purpose and scope of SPSM in the primary studies.

<table>
<thead>
<tr>
<th>Purpose and scope</th>
<th>Control and Operational Management</th>
<th>Planning</th>
<th>Process Improvement and Technology Adoption</th>
<th>Training and Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A portion of lifecycle</td>
<td>[49]</td>
<td>[99], [62]</td>
<td>[53], [63], [47], [93]</td>
<td>[91], [57], [62]</td>
</tr>
<tr>
<td>Development project</td>
<td>[75], [65], [105], [104], [85], [45]</td>
<td>[41], [45], [89], [74], [44], [94]</td>
<td>[77], [88], [65], [13], [67]</td>
<td>[60], [96]</td>
</tr>
<tr>
<td>Concurrent projects</td>
<td>[40]</td>
<td>[98]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long term evolution</td>
<td></td>
<td>[92]</td>
<td></td>
<td>[54], [106]</td>
</tr>
<tr>
<td>Long term organization</td>
<td></td>
<td>[59]</td>
<td></td>
<td>[52]</td>
</tr>
</tbody>
</table>

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

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

Table 3.15 shows a cross-analysis of purpose and simulation approaches.

SD has been used for all purposes, while an emphasis is given to planning (12 studies), followed by process improvement (6 studies) and training and learning (6 studies). Only two SD studies focused on control and operational management.

DES has been used in a balanced way for planning and process improvement, while only once SD study has been used for control and operational management. Two studies used DES for training and learning.

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 (4 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 two studies, one study focusing on planning and one on process improvement.

PN studies primarily used the simulation for planning purposes, only one study focused on process improvement.

State-based simulation was solely used for process improvement in two studies.

Others have been used in a balanced way across purposes.

3.6 Systematic 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.

3.6.1 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 1 % (5 studies) cover at least four context facets [36] and over 60 % have less than two context facets described in the articles.
Table 3.15: Purpose and approach of SPSM in the primary studies.

<table>
<thead>
<tr>
<th></th>
<th>Control and Operational Management</th>
<th>Planning</th>
<th>Process Improvement and Technology Adoption</th>
<th>Training and Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>[49]</td>
<td>[78], [99], [101], [62], [87]</td>
<td>[53], [88], [90], [63], [52], [62]</td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>[45]</td>
<td>[13], [60], [96], [51]</td>
<td>[74], [77], [60], [13], [60], [91], [57], [96], [51], [106]</td>
<td></td>
</tr>
<tr>
<td>Monte-Carlo</td>
<td></td>
<td>[61]</td>
<td>[56]</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>[75], [65], [103]</td>
<td>[75]</td>
<td>[50], [58]</td>
<td>[65], [55]</td>
</tr>
<tr>
<td>PN</td>
<td></td>
<td>[39], [97], [66], [81], [69]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>[105], [85]</td>
<td>[41], [40], [84], [93], [44], [94], [45], [72], [71], [102], [47], [93], [48], [70], [98], [100], [67], [54]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State-based</td>
<td></td>
<td>[89], [82]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.6.2 Verification and Validation of simulation models

To validate the model structure and representativeness: 18 % of the studies used “practitioner feedback” Only two studies i.e. 3 % reported performing sensitivity analysis on the model. 78 % did not report any such effort.

It can be said that in some of these studies this level of validation 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 behaviour: 51 % of the studies compared the simulation...
model output with real data. 6% reviewed the model output with practitioners. 6% either compared the model output with literature or with other models. 37% reported no such effort.

### 3.6.3 Rigor and Relevance of primary studies

Figure 3.2 provides an overview of how the primary studies in this review scored against the rigor-relevance criteria. From this figure we can see five clear sets of studies as listed below. For example, studies are classified as ‘E’ if they have a rigor score in the range of ‘0’ to and including ‘1’, and a relevance score of ‘0’.

- 17 Studies classified as ‘E’: with (Low Rigor, Low Relevance) of ([0;1], 0)
- 44 Studies classified as ‘D’: with (Low Rigor, Median Relevance) of ([0;1], 1)
- 3 Studies classified as ‘C’: with (Median Rigor, Median Relevance) of (1.5, 1)
- 3 Studies classified as ‘B’: with (Low Rigor, High Relevance) of (0.5, [1.5;2])
- 1 Study classified as ‘A’: with (High Rigor, Median Relevance) of (2.5, 1)

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

### 3.6.4 Evaluation of simulation for purpose

Only 15% of the studies actually reported some sort of evaluation of the model’s usefulness for the suggested purpose. 7% of evaluations were just comparison of predicted data to real-data (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 reports a follow-up study where the effectiveness of simulation in software process improvement is measured.

### 3.7 Discussions

*Context:* It is critical to identify the contextual factors because their interaction influences what works well in a certain setting [36, 23] highlights the importance of contextualizing empirical evidence by aptly raising the question "What works for whom, where, when and why?". Unfortunately, given that 60% of the articles describe less
Figure 3.2: Overview of Rigor-Relevance rating for the primary studies.

than two facets of context it is impossible to perform an analysis where we may es-

tablish a relation between the usefulness of simulation approaches in a given context
using the empirical evidence from the primary studies.

Evaluation goals: Given the lack of empirical data in industrial contexts one would
tend to agree with Dickman et al. [59] when they say “Methodologically, the sim-
plest 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. The reason could be that the SPSM study’s goals were aligned
with the available data or that the companies where the studies were conducted have
extensive measurement programs. The first explanation is highly likely for studies
where a phase of development process is modelled with a narrow focus e.g. main-
tenance process simulation in [53]. 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 [43, 44, 42, 41, 45, 40, 78, 79, 88, 90, 71, 56, 57] since they had a strong tradition of measurements.

Model validity: Overall in terms of V&V of the simulation models built and used in the primary studies, 29% had not reported any V&V whereas 51% had some level of V&V and 19% had reported V&V of both the model’s structure and behaviour. From the simulation literature [15, 37] 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 [61] reproducing the output correctly does not validate the underlying cause effect relations in the model.

Only 19% of the primary studies partially describe the data collection, analysis and evaluation approach. 79% of these studies do not report the study design in appropriate detail (the criteria was discussed in Section 3.4.5). Only one study explained the study design appropriately. Among the four typical threats to validity in applied research only 7% of the primary studies had a discussion of two or three threats to validity. Remaining 93% of the primary studies did not have a 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 strength of evidence in these studies. From our perspective this identifies the challenge to synthesize the evidence (if any) reported in these studies.

Purpose of simulation vs. evaluation: Of the 68 primary studies we found a diverse coverage of simulation purpose, model scope and simulation approaches. This was summarised in the Section 3.5.7 and Section 3.5.6. 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, these primary studies scored poorly on both dimensions. The only study [61] with high rigor and median relevance score did not perform evaluation of the model for the proposed purpose. Let us therefore also look at the studies in category ‘B’ and ‘C’ (that scored high on at least one or medium on both dimensions) as reported in Section 3.4.5 that also performed evaluation of simulation models for their intended purposes. We have three such ‘B’ studies and one ‘C’ study.

Class ‘C’:
Antoniol et al. [49] modelled the software maintenance process using DES to monitor and assess the likelihood of meeting a project deadline. They calibrated the model on the training dataset and then compared the predictions of the model with the remaining real data.
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

Class ‘B’:
Hsueh et al. [65] 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 game 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. [66] 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.

Houstan [63] 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 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.

Coverage of Purposes: Overall, studies taking a long-term perspective are under-represented, only four studies looked at long term evolution, and long term organization (see Table 3.14). Furthermore, there are only two 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 lifecycle. Looking at the intersection between purpose and lifecycle 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 except state-based testing, which was only applied for process improvement and technology adoption.

Impact vs. Strength of evidence: In Table 3.13 we identified the most influential articles from an academic point of view by ranking them according to citations per year. Overall, one would hypothesize that these are studies with high rigor and relevance scores among the whole set. The ten articles with the highest rigor and relevance scores (the sum of the two) are [61, 63, 49, 71, 58] with a score of 2.5 and [97, 66, 65, 43, 54] with a score of 2.0, respectively. Only two studies [49, 54] are among the top ten cited
articles that also achieved high rigor and relevance scores. From a research perspective, this implies that we need to better understand the relationship between completeness of reporting and impact. Making a mismatch explicit between those two dimensions should help us as researchers to strive for more complete reporting, and consider it as a criterion when selecting and citing work. In future studies, it is of interest to investigate whether this is a unique observation, or whether this has also been observed in other literature reviews that assessed the quality of articles.

Comparison with previous literature reviews: The total number of simulation studies (68 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. [31] 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 [1] reported on the frequency of studies with respect to domain, which partially overlaps with what we defined as 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 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 can not 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 [1] 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.

Model building cost: 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. However except the studies from Dietmar Pfahl none of the included studies reported the effort spent on simulation. Pfahl and Lebsanft [85] 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 [84] only reports the calendar time of three months for knowledge elicitation and modelling and four meetings that were held in this period between Fraunhofer IESE and Daimler Chrysler.

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.
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

Assessing the evolution of the field: In order to evaluate how the field of process simulation evolved over time, it is very important to understand how different studies depend on each other, and which additional value they are reporting. We identified cases of journal articles that were extended from conference articles do not explicitly acknowledge that these are extension of previously published work thus making it almost impossible to identify such studies, e.g. see the pairs [68] and [39], [60] and [7], [5] and [58], [6] and [100], [72] and [9], as well as [88] and [16].

Furthermore, some studies are remarkably similar, e.g. see the pairs [45] and [8], [4] and [53], as well as [106] and [38]. A methodology to develop simulation model proposed by Park et al. is reported in three articles which are very similar as well [13, 3, 2]. 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.

There is some trend of multiple publications reusing the same model and results in the SPSM area as well. For example, Abdel Hamid developed a SD model for project dynamics and validated it by reproducing the results of a project at NASA. Then this model was used as a vehicle for experimenting with that model and results were reported in six different articles [43, 44, 42, 41, 45, 40].

Similarly, Zhang et al. have reported a two phase systematic review in five articles [27, 28, 29, 30, 31]. There is an overlap in contribution of at least three of these articles [27, 28, 29]. In these particular cases the research community would be helped by reporting those studies in a more aggregated fashion, e.g. in a single and more extensive journal publication.

3.8 Conclusions

In this study we aggregated the literature of software process simulation applied in real-world settings. We identified a total of 68 articles and for increasing the validity of the results the authors individually assessed the articles and discussed in case of disagreement. In a first step we mapped the articles to identify which purposes they have been used for, which scope they were focusing on, and what are the articles and authors with the highest impact.

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

With regard to the mapping 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. Concurrent projects are of particular interest in case of large scale software development where several teams work on a software system.

Figure 3.3 provides an overview of achieved validation and evaluation of usefulness. The figure shows that 19% of the included studies verified the model structure and behaviour. Overall, 18% provided an evaluation. However, the evaluation done was at best only static according to the definition from [13], i.e. feedback from practitioners were 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, and at the same time conducting an evaluation against specified purpose.

Figure 3.3: State of model validation and evaluation.

With respect to the evaluation of rigor and relevance in relation to what was reported in the studies, the majority of studies achieved scores below median rigor and relevance values.

For practice and research we provide the following recommendations:

1. When 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.

2. From a research perspective, future studies should not just focus on replicating reference behavior, as many studies have shown that this was successfully
Chapter 3. Software process simulation: A systematic review on rigor, relevance, and impact

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 context in industrial studies [36], Runeson and Höst [17] for case study research, Wohlin et al. [18] on conducting and reporting experiments. For reporting SPSM studies using continuous simulation a good template is available in Madachy’s book [15].

We also evaluated the relation between impact and quality of reporting in relation to rigor and relevance, and found that only two out of 10 articles with the highest rigor and relevance scores were among the most influential studies, indicating a weak relation between reporting/study quality and academic impact.

None of the existing systematic reviews [32] and [27] [28] [29] [30] report this lack of evaluation in SPSM research. Zhang et al. [31] do point this out but do not provide any traceable evidence to conclusively show the need for evaluation in SPSM.

Overall, concrete future studies can be recommended based on this review. From a simulation perspective we need complete evaluations and reporting sufficient information to achieve high rigor and relevance scores. Furthermore, in general it is of interest to understand the relationship between study quality and impact, as making this relation explicit opens up an opportunity to further improve empirical research.

Recently there is an indication that researchers from the SPSM community are trying to ponder on reasons why software process simulation has not had the impact it ought to. Based on the results of this review, we can see that it will be difficult to build a case for SPSM in industrial use without reporting the evidence for its usefulness and the cost of adoption.

It is well established now that different simulation approaches “can be” used to simulate 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.
3.9 References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES
Chapter 4

Testing highly complex system of systems: An industrial case study

4.1 Abstract

Systems of systems (SoS) are highly complex and are integrated on multiple levels (unit, component, system, system of systems). Many of the characteristics of SoS (such as operational and managerial independence, integration of system into system of systems, SoS comprised of complex systems) make their development and testing challenging. This study provides an understanding of SoS testing in large-scale industry settings with respect to challenges and how to address them. The research method used is case study research. As data collection methods we used interviews, documentation, and fault slippage data. We identified challenges related to SoS with respect to fault slippage, test turn-around time, and test maintainability. We also classified the testing challenges to general testing challenges, challenges amplified by SoS, and challenges that are SoS specific. Interestingly, the interviewees agreed on the challenges, even though we sampled them with diversity in mind, which meant that the number of interviews conducted was sufficient to answer our research questions. We also identified solution proposals to the challenges that were categorized under four classes of developer quality assurance, function test, testing in all levels, and requirements engineering and communication. We conclude that although over half of the challenges
we identified can be categorized as general testing challenges still SoS systems have their unique and amplified challenges stemming from SoS characteristics. Furthermore, it was found that interviews and fault slippage data indicated that different areas in the software process should be improved, which indicates that using only one of these methods would have led to an incomplete picture of the challenges in the case company.

4.2 Introduction

System of systems (SoS) recently received vast attention in the software engineering research literature. A system of systems is characterized through operational and managerial independence in the development of the individual systems that should later on act together, and is characterized by an integration of many different systems into a new system. System of systems are also generally very complex, and there exist suppliers that deliver them for integration [10, 12]. Literature (cf. [6, 9]) distinguishes different types of SoS, namely virtual, collaborative, acknowledged, and directed SoS.

It is acknowledged that SoS development and quality assurance is very challenging, e.g. due to involvement of many parties, it is not easy to integrate systems continuously, and so forth [6, 10, 11, 12]. However, so far there is a lack of empirical studies that explore the challenges and possible solutions of how to test such complex system of systems. In response to this research gap this case study makes the contribution to investigate the challenges and potential solutions of SoS development in an industrial case study of a large-scale system of systems from the telecommunication domain with over 5,000,000 Lines of code. The following contributions are made by this study:

- Understand how SoS testing is done by describing and characterizing how system of systems are currently tested based on a case having the typical characteristics of directed SoS development.
- Identify the challenges of SoS testing observed in the case and categorize them to three classes: a) challenges that are not different in SoS context and other contexts, b) challenges that are amplified in SoS context but that can also be found in other contexts, and c) new challenges specific only to SoS context.
- Identify possible solutions of SoS testing based on the challenges.

The remainder of the chapter is structured as follows: Section 4.3 presents the related work. Section 4.4 describes the research method, followed by the results in Section 4.5. Section 4.6 concludes the papers by presenting the observations and implications given by the results.
4.3 Related work

The Systems Engineering Guide [13] defines SoS “set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities”. Maier [6] and Dahman et al. [9] classify SoS into Virtual, Collaborative, Acknowledged, and Directed. Virtual means that there is no central management or agreed purpose of the SoS, purposes emerge as systems are combined. This reflects the philosophy of service-oriented architecture, which is a way of implementing SoS [11]. Collaborative SoS have an agreed purpose and interact voluntarily. How systems should interact is collectively decided between system owners. Acknowledged SoS are characterized by common objectives, and there exists a designated manager that assures resources for SoS development. The System Engineering Guide stresses that the individual systems still have independent ownership, objectives, funding, and development. Directed SoS are built to fulfil specific purposes and they are managed around those purposes. There exist a central management for the SoS. However, the system development organizations contributing individual systems are still independent, but subordinate to the purpose. The SoS in this study is characterized as directed.

The System Engineering Guide [13] also highlights some engineering challenges that are particular to SoS, namely (1) Management issues with respect to governance of system development across organizational boundaries, (2) the increased complexity, scope, and cost of processes for planning and engineering, (3) achieving and maintaining interoperability, and (4) likelihood of unpredicted behaviour, and hence the lack of predictability of behaviour.

Lewis et al. [10] point out that SoS challenges are related to collaboration, in particular who collaborates and what everyone has to provide so that the collaboration would work. Furthermore, there are particular challenges if systems do not evolve in a similar pace or manner, which makes their integration challenging. Hence, if there are some dependencies (one system cannot fulfil its service without another) then there is a need for synchronization, which makes collaboration essential. How the SoS is built in an enterprise affects the severity of the challenge of collaboration. They distinguish SoS in an enterprise where a single or multiple organizations develop, and where multiple enterprises develop.

Columbi et al. [12] characterizes SoS as systems that operate synergistically, however, SoS can operate and be managed independently. They acknowledge and highlight that these characteristics make testing and evolution of SoS very challenging. In other words, they have amplifying character, or even lead to completely novel challenges. In their experience report from a SoS development at the Department of Defense (DoD), they suggest a number of challenges: (1) organizational structures do not support test-
Chapter 4. Testing highly complex system of systems: An industrial case study

ing in a SoS framework; (2) No steps in testing that evaluate the overall SoS capabilities, rather the focus is very much on individual systems (no obligation for the project manager to scope testing on the SoS level), (3) Rate of incoming requirements has drastically increased on the SoS, leading to an exponential increase of test events, complexity, and expense with respect to testing; (4) No overarching test scenarios across different systems due to coordination and communication breakdowns. In response to the challenges, they provided a number of solutions: (1) only test interoperability with respect to changes made, and do not test everything, but rather "built a little, test a little", (2) use risk assessment to prioritize the ever growing number of tests, (3) focus tests on interfaces as there the greatest risk of mistakes lie, but do not neglect internal behaviour of systems, (4) provide designated environments and people for integration.

Furthermore, we identified other studies that have a focus on SoS quality assurance (see e.g. [8, 7]), but those are solution proposals without an industrial partner involved.

Overall, we found that there is a lack of systematic empirical evidence that investigates how to conduct testing in a SoS context, and relates that to challenges and solutions.

4.4 Research method

As a research method we used case study research, following the guidelines provided in Yin [16] as well as Runeson and Höst [5].

4.4.1 Purpose and research questions

The purpose of the study is to gain an in-depth understanding of testing practices, challenges, and potential solutions when testing very large SoS.

**RQ1: How is SoS testing done in industry?** We first need to gain a deep understanding and rich description of the current situation, as SoS testing is not well described in literature. The case being studied in that sense provides an interesting case as it fulfils the characteristics of directed SoS development very well.

**RQ2: What are perceived and measured challenges when testing SoS and how are they different from testing challenges of other contexts?** Given testing in SoS is not well explored in industry the first step should be to understand the challenges in order to find useful solutions.

**RQ3: What potential solutions do practitioners see in order to address the challenges identified?** The identification of the solutions is based on experiences made in the SoS context. They provide useful input for future evaluations to test their actual impact in the studied context.
4.4.2 Case and context

The case being studied is a development site of a large Telecom vendor, engaging in SoS development. The case and context are described, as this allows for generalizing the results to a specific context. Other companies in a similar context are likely to find the results transferable to their context [14].

The process used at the company follows a SoS approach. There is no common definition of system of systems, as the term has been defined in different domains, such as military, enterprise information systems, or education [15]. The term has been recently established in the software engineering field, where a system of systems should fulfil several of the characteristics. These characteristics and their presence in the case company are shown in Table 4.1.

Table 4.1: System of Systems Approach Characteristics (cf. [15]).

<table>
<thead>
<tr>
<th>ID.</th>
<th>Characteristic</th>
<th>Case Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Operational independence</td>
<td>(√)</td>
</tr>
<tr>
<td>C2</td>
<td>Managerial independence</td>
<td>(√)</td>
</tr>
<tr>
<td>C3</td>
<td>Integration of system into system of systems</td>
<td>(√)</td>
</tr>
<tr>
<td>C4</td>
<td>SoS comprised of complex systems</td>
<td>(√)</td>
</tr>
<tr>
<td>C5</td>
<td>System suppliers deliver systems for integration</td>
<td>(√)</td>
</tr>
<tr>
<td>C6</td>
<td>Complete technical overview of SoS and system supply</td>
<td>(√)</td>
</tr>
</tbody>
</table>

The overall architecture of the system of systems studies 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 4.2. In the first step the high-level requirements for the overall SoS 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 4.2). Each system can be integrated independently of another system, which provides them some degree of operational and managerial independence (see Table 4.1). However, the versions of two systems have to be compatible when the system of systems is 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. This fulfils the
characteristics of SoS development related to system complexity and integration. In order to make sure that the system of systems is working together in the end an overall system structure and design is developed, referred to as the anatomy. This allows having an overview of the overall SoS, also making explicit how each system in the SoS contributes to the overall system goals.

Looking at other context elements [14] the following should be added as information:

- All systems and the SoS are older than 5 years.
- 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

4.4.3 Data collection

We used multiple data sources for triangulation to increase the validity of the results. In order to answer RQ1 and RQ2 we used documentation, interviews as well as quantitative data as input. For RQ3 we used interviews to gather opinions of how to best improve SoS testing.

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 is 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 before conducting the interviews. In addition we consulted official templates for reporting (e.g. forms for documenting defect data).

Interviews It is important to point out that the sampling procedure here is not to get a representative sample of a population, but rather having a diversity of perspectives to be able to contrast alternatives in the evaluation step. We notice that with each additional interview there was a high overlap in the views and we decided to stop after five interviews. Even though the practitioners had diverse roles the number of challenges and improvement suggestions stabilised after the third interview, which indicates a good coverage of the interviews (refer to Figure 4.1).

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. In the following a short profile of the testers is provided in Table 4.2.

**Interview structure:** The interview is structured in four themes, namely: (1) interviewee knowledge and experience, (2) when and where to detect what type of defects, (3) how the tester conducts tests, and (4) strengths and liabilities of the current testing approach. The interview is semi-structured and consists of mainly open-ended questions, hence allowing to follow and discuss interesting issues, or change the order of questions. The interviews approximately took 90 minutes to complete. The contents of the interview were as follows:

- **Part I: Interviewee knowledge and experience:** This theme focuses on obtaining information about the testers knowledge and practical experience (professional background and education, explaining current role at the company, roles before current role).
- **Part II: Understanding test levels and their responsibilities:** This theme focuses on determining which types of defects are detected at which test level, using a defect classification scheme for telecommunication systems from Damm and Lundberg [18].
- **Part III: Understanding how testing is done and its related challenges:** This theme focuses on how testing is done and what challenges occurred, including characterization of test objects at different levels, input used to derive test cases and the quality of the input used (e.g. requirements), tool support, and general strengths and challenges observed.
Table 4.2: Practitioners’ profile.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The interviewee is currently working with test improvement with a particular focus on FT and ST, and has experience from working in testing teams before. Furthermore, the interviewee has more than five years of experience focused on testing.</td>
</tr>
<tr>
<td>2</td>
<td>The interviewee is working in software development since 1995 and is currently a technical coordinator. In the past the interviewee also worked as a tester, and had a leading role in design teams with responsibility for product and test code.</td>
</tr>
<tr>
<td>3</td>
<td>The interviewee is a technical expert assisting software developers in solving technical problems, in particular the problems that occurred in testing. Hence, the interviewee is directly exposed to review and BT in the daily work.</td>
</tr>
<tr>
<td>4</td>
<td>The interviewee is working with the company since 1994, and has been involved in software testing, development, support, and system management. She has worked as a developer for several systems in the company, currently leading a development team investigating new enhancements to the main system. The interviewee has experience of conducting reviews, BT, and FT.</td>
</tr>
<tr>
<td>5</td>
<td>The interviewee is currently working in ST and CST. Before that the interviewee had various technical roles, including support and software design. Overall, she has over 10 years of experience in designing software.</td>
</tr>
</tbody>
</table>

**Part IV: Closing:** The interviewees were asked whether they want to add something important that was not covered in the interview questions yet.

Before starting the actual interview, the purpose of the study and the reasons for the selection of the interviewee were provided. Furthermore, the interviewee was asked whether we were allowed to record the interview for transcription. The interviewee was also informed that all information collected was treated anonymously and will be aggregated with the information provided by the other interviewees.

*Quantitative data:* With respect to defects we looked at distribution of types of defects and defect criticality per test level, and fault slippage. The types of defects discovered at the test levels reveal which types of defects the test level is actually able to capture given the current test practices employed. Fault slippage measures indicate whether each testing phase is able to detect the defects it is supposed to detect (see [19]). The data is available through a company internal and proprietary defect reporting system. For this study we focus on recent defects (in the past 12 months) that reflect
the current test strategy employed at the company.

### 4.4.4 Data analysis

*Interviews:* Audio recordings of all five interviews were transcribed. We used colour coding for initial data extraction from transcribed interviews where one unique colour was assigned to each one of the following:

- Challenge, problem, malpractice, limitation and missing information for testing
- Benefit or strength of current practices, tools or processes
- Current practice, way of doing work or tools used
- Improvement suggestions
- Definitions of terms, test levels and artefact descriptions

While colour coding, brief notes were made about the statements making use of the context of the statements and reducing misinterpretation later on. These colour coded statements and their brief descriptors were extracted and were put in a spread-sheet verbatim while maintaining traceability to the source. At this point these were assigned codes (according to Table 4.3) to capture their relation to the respective test level. A separate spread-sheet was created for each of the five items above. Next step was to aggregate the repeating statements which was done by repeating the following this process for all five sheets:

**Step-1:** For the first statement create and log a code.

**Step-2:** For each subsequent statement identify if a similar statement already exists. If it does log the statement with the same code, otherwise create a new code.

**Step-3:** Repeat Step-2 until the last statement has been catalogued.

A short description was given to each of the resulting clusters with same code. As traceability was ensured between the clusters, their summary/description, individual statements and audio recordings, second author was able to review the results of the process whether the statements were correctly clustered together. The disagreements were resolved by discussion.

### 4.4.5 Threats to validity

While designing and conducting this case study various conscious decisions were taken to strengthen the validity of results. Using the checklist proposed by [5] we evaluated the case study protocol. This ensured that we had addressed all the critical requirements of case study design including aims of the study, defining the case, unit of analysis and
data collection methods among others. Another researcher who has extensive knowledge and experience in case study research also reviewed the protocol. Furthermore, this detailed protocol was kept up-to-date, reflecting the actual course of the case study.

**Construct validity:** Both methodological (interviews and archival data analysis) and data source (practitioners and defect database) triangulation were used to strengthen the evidence generated in this case study. Using appropriate amount of raw data and through clear chain of evidence (maintaining traceability between the results, qualitative data and sources), this validity threat was minimized.

**Internal validity:** In this case study, the challenges were examined in the context of SoS, and the relation of SoS context to challenges was discussed. However, given the complexity of a real world organization and the fact that confounding factors are always a challenge when studying real world systems, the isolated effect of SoS characteristics can not be established, and requires further investigation.

**External validity:** There are too few empirical studies yet to make general claims, but the case in this study represents a typical complex SoS based software product development situation and is likely to apply to similar contexts w.r.t. system complexity, domain, etc.

**Reliability:** By involving more than one researcher, each actively engaged in the design, review and execution of the study, where case study protocol was the means of communication and documenting the agreement. Thus we believe that the resulting detailed protocol serves as a good means to replicate this study. The detailed documented steps of data collection, processing and analysis also increase the reliability of this study.

## 4.5 Results

The results present the current practices of testing SoS, the challenges and their relation to SoS characteristics, as well as SoS solution proposals.

### 4.5.1 SoS testing process and practices (RQ1)

*Test process:* We found that overall there are six Test levels at the case company. These were visible in both the test process documentation in the company and the interview results. Furthermore, the organization of testing teams and responsibilities is roughly organized around these levels as well. We found additional test levels in the defect database, however, based on the defect reports from the last two years these levels were not used at all in reporting. This hints redundancy of some of these levels. One
explanation for no faults reported to “Regression test” is that regression testing is performed within other phases of testing and therefore is not considered a separate phase. Based on the congruence between the latest documents, interview results and defect reports we found six levels as shown in Table 4.3. These test levels are depicted in the overall development process in the Figure 4.2.

Table 4.3: Test levels in the case company.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description of test levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev</td>
<td>Reviews using static analysis and Visual “Diff” tools are performed by the development teams.</td>
</tr>
<tr>
<td>BT</td>
<td>Basic test is done in the Agile Team (AT) sprint by the development teams and uses implementation as the test reference. It includes unit testing with JUnit [24] and TestNG [26]. Input for tests are user stories, protocol specifications, component descriptions, dimensioning requirements, and traffic models.</td>
</tr>
<tr>
<td>FT</td>
<td>Function test focuses on testing isolated functions and is done in AT sprint by the development teams using Testing and Test Control Notation (TTCN) [25].</td>
</tr>
<tr>
<td>ST</td>
<td>System test tests the integration of the AT sprints (teams) in pre-defined cycles. Test cases created in FT and previous releases are used. No new test cases are written. The testing is done on a different branch along with regression testing using self-contained test cases (each test case sets up configuration, executes test scenario, and removes configuration once it is done). In addition, the company uses traffic models and simulates the system behaviour under load.</td>
</tr>
<tr>
<td>CST</td>
<td>Compound system test, tests the integration of systems into the overall system. Here, the emphasis is on making sure that the system functionality when integrated, still fulfils non-functional requirements (e.g. load), and is installable.</td>
</tr>
<tr>
<td>FOA</td>
<td>After system test is completed, the product is tested by installing it on a trial customer network. After this the product is launched to the general market.</td>
</tr>
</tbody>
</table>

Test levels and their responsibilities: We used interviews to explore the practitioners’ perception as to which defects are found at each of these test levels. The results are presented in Table 4.4. The first column depicts the fault classification used in the case company and the numbers in each row show how many of the five practitioners interviewed consider that this defect type is often found at this level. From the opinions expressed in the interviews most of the defects are found at ST. For certain defect types this is an expected behaviour e.g. quality assessment at system level is not possible
before this level. Perhaps that is why there is a consensus that Quality related issues (performance, robustness and concurrency) are found in ST. One practitioner said, “the single biggest place where we find faults is in ST”. Practitioners perceive that programming faults are discovered in basic testing which is an expected behaviour. Although reviews are recognised as a good practice and are used in the company but in practitioners’ opinion it is not effective in discovering defects. Only one practitioner was satisfied with the effectiveness of current reviews.

*Test efficiency in the current situation:* As a measure of test efficiency the com-
Table 4.4: Fault classes [18] and where practitioners find them based on interviews.

<table>
<thead>
<tr>
<th>Fault class</th>
<th>Rev</th>
<th>BT</th>
<th>FT</th>
<th>ST</th>
<th>CST</th>
<th>FOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Interface</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Interface</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human Interface</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coding Errors / Coverage</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redundancy</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concurrency</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing Functionality</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong Behaviour</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

pany uses fault slippage, indicating whether a fault is found in a later test level than the one where it ought to be found. From Table 4.4 we observed that there is some fault slippage in the testing process. For example, consider the defects triggered from “internal interfaces” and “Coding errors” that should ideally be found in the early test levels like reviews or basic test. However, in practitioner’s opinion these defects often slip through to later stages.

To assess the validity of these observations and look at the effectiveness of testing quantitatively we used the FST data. We analysed the defect reports for the last calendar year since 01.01.2011- 01.02.2012 (see Table 4.5 for the results).

Table 4.5: Fault slip through data.

<table>
<thead>
<tr>
<th>‘Should’ to ‘Did’ detect</th>
<th>Rev</th>
<th>BT</th>
<th>FT</th>
<th>ST</th>
<th>CST</th>
<th>FOA</th>
<th>Total slippage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev</td>
<td>30.76</td>
<td>7.69</td>
<td>30.76</td>
<td>19.23</td>
<td>0.00</td>
<td>11.53</td>
<td>25.71</td>
</tr>
<tr>
<td>BT</td>
<td>6.15</td>
<td>26.15</td>
<td>32.30</td>
<td>18.46</td>
<td>4.61</td>
<td>12.30</td>
<td>62.85</td>
</tr>
<tr>
<td>FT</td>
<td>7.14</td>
<td>50.00</td>
<td>28.57</td>
<td>14.28</td>
<td></td>
<td></td>
<td>8.57</td>
</tr>
<tr>
<td>ST</td>
<td>88.88</td>
<td>11.11</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CST</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>

In Table 4.5 the values on the diagonal (in bold face) represent the percentage of defects that were found exactly where they are ought to be found. The percentage points in the last column “Total slippage” accounts for the percentage of slippage from each
level compared to the total slippage from all test levels. In terms of overall slippage from one test level Basic test (with 62.85%) and Reviews (with 25.71%) have a lot of potential for improvement. From these results ST and CST are the top performers in terms of catching the right faults. As pointed out by one of the practitioners, “The single biggest place where we find faults is in function test” we can see that most of the slippage from Reviews and BT are caught in Function test (30.76 % and 32.30% respectively).

In Table 4.5 there are some unexpected values below the diagonal (in italics). There could be a number of explanations for it, the simplest one that it was a mistake in reporting the data or that although the test strategy states that the defect should have been found later but it was found in an earlier phase, thus requiring an update to the strategy or it was just a coincidence that this defect was found earlier and doesn’t warrant a strategy update. However, we did not confirm or reject any of the explanations for this small percentage of defect reports.

We can see some congruence between results from Table 4.4 and Table 4.5. The majority of the defects are found in the later test levels especially in FT and ST. Top two levels with maximum slippage are BT and reviews respectively.

Table 4.6: Fault slip through to customer.

<table>
<thead>
<tr>
<th>‘Should’ to ‘Did’ detect</th>
<th>Percentage found by Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev</td>
<td>15.27</td>
</tr>
<tr>
<td>BT</td>
<td>63.88</td>
</tr>
<tr>
<td>FT</td>
<td>6.94</td>
</tr>
<tr>
<td>ST</td>
<td>11.11</td>
</tr>
<tr>
<td>CST</td>
<td>1.85</td>
</tr>
<tr>
<td>FOA</td>
<td>0.00</td>
</tr>
<tr>
<td>Customer</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Looking at the problems found by the customer (as shown in Table 4.6) we can see that a high percentage of slip through is from Basic testing and Reviews. Having triangulated the practitioners’ observations with quantitative data, we delved deeper to understand the practical challenges and their influence on these symptoms.

4.5.2 Challenges (RQ2)

From the interviews and defect databases we found three main undesired results with respect to outcome variables in testing, namely fault slippage, long turnaround time
and maintenance of test suite. These outcomes are influenced by various issues and challenges, which were also identified from the interviews. In total we identified 30 different challenges that were mapped to three levels: 1) challenges that are not different in SoS context and other contexts, 2) challenges that are amplified in SoS context but that can also be found in other contexts, and 3) to new challenges specific only to SoS context (see Figures 4.3, 4.4 and 4.5). We do not claim cause-effect relations, as those would have to be established in controlled environments. Hence, the relation between challenges and outcome variables should be seen as indicators in this exploratory study. The challenges in levels “2” and “3” as described above are also mapped to the characteristics of SOS (presented in Table 4.1) that aggravate or cause these challenges.

Figure 4.3, shows the challenges that influence the undesired result of fault slip-page. In the related work (Section 4.3) it was highlighted that the independence of system development leads to challenges in collaboration. As there are many organizations integrating their interacting system and there are many levels of testing (see C1, C2, and C3 in Table 4.1), the challenge of unclear responsibilities of test levels is apparent. Collaboration challenges and independence (C1 and C2) also affect knowledge sharing, and hence lack of compliance to processes, lack of shared guidelines for testing at various levels and tools. It also leads to a lack of responsibility for test suits that are shared by different systems when they are integrated. SoS affects testability of requirements due to its complexity (C4), and that a requirement concerns multiple systems and in order to interpret them reasonably well a wide domain expertise across system boundaries is needed. There is also a lack of thorough analysis and selection of test cases, which is influenced by the SoS complexity (C4). Other influencing factors (e.g. early evaluation of quality attributes [4], time to market pressure and influence on test [23], difficulties in basic test, and lack of independent verification (people who code also write the test cases) [3] are rather generic challenges.

In Figure 4.4 the challenges to the maintenance of regression suite are visualized. Because of the growing number, size and complexity of systems (C4) put together in the SoS (C3 and C5) it is imperative that the system test will get more and more test cases related to FT in regression suite. Furthermore, there is no formal responsibility for maintenance of the test suite given characteristics C1 and C2, which also leads to no detection of redundant and obsolete test cases. The sheer amount of test cases added makes it a difficult problem to address. Any centralized solution is likely to be overwhelmed and with no shared standards (difficult to achieve in a SoS with managerial and operation independence) the current decentralized approach is not working, as the quality of test cases in terms of design, implementation and readability is a major issue for maintenance. Challenges such as test case readability (e.g. addressed to behaviour driven development [2]), company proprietary code due to lack of tool awareness, and lack of separation between test code and test data [21] are rather generic. However,
Chapter 4. Testing highly complex system of systems: An industrial case study

Figure 4.3: Challenges influencing fault slippage in the case company.

non-compliance is aggravated in SoS due to difficulties in spreading news about good tools.

Figure 4.4: Challenges influencing maintenance of FT regression suite.

The challenges contributing to a long turnaround time for regression test are shown in Figure 4.5. The number of growing number of FT test cases affects the maintain-
ability of the test suite, and is significant due to the SoS context, as was discussed for Maintenance. The difficulty of doing basic testing leads to the misuse of the FT framework, which is amplified by the SoS complexity (C4) and lack of communication and interaction (C1 and C2) as the framework now includes many BT tests, that take more time to be executed due to the limitations of the FT framework. It should also be emphasized that many of these challenges influence multiple problems and have some interactions between them as well. The platforms inefficiency is one dimension of the problem because it is not easy to switch to a different framework because of the cost of migration of existing test cases and retraining the resources. Therefore the solutions to improve the turnaround time may take a multi pronged approach e.g. reducing the number of test cases to run by prioritization, selection or using the framework for its strengths. Another option is to think about smarter ways to design to specifically improve the turnaround time. Prioritization of test cases [20, 1] and TTCN as a test framework [22] have been addressed generically in research contributions.

Figure 4.5: Challenges influencing the turnaround time of FT regression suite.

4.5.3 Improvements (RQ3)

We identified 14 improvement actions in order to reduce/ mitigate the negative impact of the issues on fault slippage, maintainability of test cases, and turn-around time. The improvement suggestions have been organized to five groups based on which area they affect. The groups are developer quality assurance, function test, testing at all levels, requirements engineering and communication.
Table 4.7: Improvement ideas and effected testing processes.

<table>
<thead>
<tr>
<th>Improvement ideas</th>
<th>Rev</th>
<th>BT</th>
<th>FT</th>
<th>ST</th>
<th>CST</th>
<th>FOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reviews</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Testing</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance of FT reg suite</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controlling size of FT reg suite</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback time for FT reg suite</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidelines for FT tool usage</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tech improvements to FT tool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Improving test case quality</td>
<td></td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Definitions of test levels</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Increase tool usage:</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Early quality evaluations</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Requirement testability</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Feature status tracking</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Improved interaction</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Developer quality assurance

The improvement suggestions in this group affect the work practices of individual developers and the results produced are not used outside of individual AT.

Reviews: Code reviews have helped in early detection of coding fault, however, there is a need for organization wide adoption of the practice. Reviews should be supplemented with tools that support visualization of code changes in the artefact under review. To utilize the human resources effectively, automated static code analysis should precede reviews. This will enable the reviewers to focus on the defects that the tools cannot find. These tools are generally easy to use and if configured properly can reduce the number of false positives. Thus, reviews can help avoid certain fault slippage and provide means for early evaluation.

Basic testing: Practitioners from development teams believe that increased basic testing has reduced the number of faults that slip through to the function test. They acknowledge that these improvements in basic test have been possible because of inclusive discussions with various development teams. This is a clear indication of why it is necessary to have stakeholder involvement and buy-in/consensus on decisions. Furthermore, some teams set and enforce coverage goals. They use code coverage tools, as these tools help to see the extent of testing. However, there is a need for wider adoption of these tools throughout the company. Some practitioners also suggested that test-driven development and use of mock object techniques could offer avenues
for improvement.

**Function test**

The improvement suggestions in this group focus on the function test level.

*Maintenance of FT regression test suite:* There was a consensus on having persons with formal responsibility for maintenance of the test suite. This seems plausible, but considering the size of the organization and amount of test cases it will require a lot of skills, effort and coordination between teams. However, the current practice of having a distributed mechanism has not really worked either. Perhaps in the SoS context the only viable solution is that of shared responsibility with some central governance. Thus, having multiple levels of control on what is contributed to the regression suite. For example, first the individual teams assess the quality of FT and suggests an inclusion to the suite, and then the tests are only included if the person(s) with formal responsibility approves them. Furthermore, improving the readability of test cases and enforcing design principles when developing test code will reduce the maintenance cost as well.

*Controlling the size of FT regression suite:* By improving the effectiveness of reviews and BT (see suggestions in Section 4.5.3) the unnecessary load on FT will be reduced. Similarly having responsible person(s) for test suite maintenance will give an opportunity to have a filter on new test cases being added. Similarly such person(s) may also review existing test cases and remove redundant or obsolete test cases from the suite, thus reducing the number of test cases.

*Reducing the feedback time for FT regression suite:* Practitioners believe that having prioritization or selection criteria will bring significant improvement in feedback time for the regression suite. So rather than running all the test cases all the time, two suggestions for prioritization criteria were: to run the most relevant tests first or to run the tests that fail frequently before others in the suite. Another step in the right direction is that the test cases have been tagged at a high-level based on the protocols and features they test. This is useful to select relevant test cases and to identify which parts of the system are being neglected in testing and which other parts have a high overlap in test cases. Removal of redundant and obsolete test cases and using the framework for FT (instead of BT) will also alleviate the problems aggravated by having a large regression suite in ST where all the FTs are added, as well as the problems related to an inefficient testing framework.

*Guidelines for FT tool usage:* We found that TTCN, i.e. the tool used in FT, is sometimes misused for unit testing which is not the strength of the framework. In large organizations dealing with SoS there is a Lock-in with respect to tools and it is very costly to switch to a different tool, especially when it is as embedded as TTCN. Therefore, the practitioners suggested to use TTCN for FT only as it has following
abilities:

- Ability to test multiple protocols: is a major strength of TTCN over other testing frameworks.
- Automation helps save time: with use of TTCN for testing the time taken for test has reduced considerably. As one practitioner said, “For us, for instance if we want to test a Product Customization, with say 30 test cases. Just executing them may be takes about 15 minutes. But doing it manually will take may be 3-4 hours”.
- Useful for protocol verification: the ability to manipulate at bit level gives better control of protocol messages and parameter verification. This makes it very useful for external interface testing.

Thus, by taking steps in Section 4.5.3 to manage the suite size we can use the resource intensive TTCN for the testing that cannot be tested without it.

Technical improvements to FT tool: Many of the maintenance problems of the FT suite stemmed from the technical shortcomings of the TTCN tool used in the company, thus various improvement ideas for it were suggested in the interviews. Practitioners had complaints about the inability of TTCN to support load and GUI testing. There is also a need to improve the TTCN editor to identify multiple test headers in the same file and improve the readability of test cases. It should be easy to write TTCN test cases without having to repeat a lot of things, and to write configurations in the test cases so that verification is facilitated. One suggestion was to develop wrappers or high-level language constructs for setting up complex configurations, which will reduce the requirements on knowledge about subsystems and intricate details of how the protocols work. This will not only make writing test cases easier, which means even under time constraints it is still possible to develop test cases without the threat of mistakes in configuration.

Testing in all levels

The improvement suggestions in this section affect testing at all levels of testing from basic testing (BT) to field tests (FOA).

Improving the test-case quality: Having templates and guidelines for writing test cases particular to various test levels and tools used will help improve the quality of test-cases. This will enforce a consistent structure in test-cases and help teams following best practices to enabling readability, good design and high coverage. Furthermore, improving the quality of test cases will significantly improve the confidence in the testing suite’s ability to find faults. The current mechanism of writing configurations at
the bit level results in test cases that are error-prone. In the complex SoS context it is almost impossible to have such detailed knowledge about all the systems. Thus, having a wrapper (which also addresses the technical issues of the FT tool) to provide abstraction for the low level configurations will facilitate writing test cases and reduce the necessity of in-depth domain knowledge about various systems in SoS. Having test-case templates, documented best practices, and high-level constructs to specify test cases will also alleviate the effects of employee turnover as the knowledge will be embedded in the artefacts. Furthermore, reducing fault slippage by writing tests in pairs and/or review test cases by another team member will reduce chances of misinterpretation of requirements.

Definitions and responsibilities of test levels: Practitioners identified the need for better organization wide definitions of test levels with clear strategies of what should be verified at each test level. They also think that a better analysis of what we want to verify in each test level can reduce the overlap in testing. It is important to make a trade-off between early detection and cost of set-up. One may identify certain faults in basic test but the cost of setting up the environment to test may be too high. It is an encouraging sign that the testers understand this e.g. while commenting on the role of basic testing one practitioner said, “We are aware that we can’t test everything and may be we shouldn’t in basic test”.

Increase tool usage: Practitioners were aware of many improvements that could be gained by wider usage of test tools. For example, generating tests instead of manual coding, while talking about the benefits of a tool used to create trees and simulation test cases the practitioner said, “then you can run a lot of logic in the tree without writing in TTCN”. Similarly, use Mock tools as suggested in Section 4.5.3 instead of statically programming the interfaces. Also, it was suggested to use tools to generate test data that helps in e.g. boundary value analysis, thus maintaining a differentiation between test code and data.

One practitioner highlighted the importance of tools that can provide a management view by summarizing and presenting data to support decision-making. This aligns well with the need of technical overview in a SoS. SONAR [27] is one such tool that can help providing:

- a management view of measurements and status.
- feedback on test cases, where there are less test cases or too many.
- and highlight, which test cases are not designed properly.

Early quality evaluations: The practitioners expressed the need to run non-functional tests sooner than currently possible. One solution is to look for performance issues that can be identified earlier without running the non-functional tests. This could be done
by reviews or static code analysis to e.g. design and complexity of algorithms, memory management that is likely to result in performance degradation. Yet another complementing solution is to have early architectural evaluations for quality requirements e.g. by using prototyping.

Requirements engineering and communication

The large size of the SoS context can easily lead to communication gaps. Furthermore, requirements form the bases against which tests are developed. Thus, it is natural that practitioners suggested how improvements in requirements engineering and communication would help the case company.

Improving the testability of requirements: In a typical case in SoS context, requirements are written to describe a system service that involves multiple systems. It therefore helps when the feature lead (person owning a feature) does a pre-analysis of requirements to understand which systems will be affected and helps testing teams to anticipate workload. Similarly, having concrete user-stories that describe one standard scenario and not the whole service is useful for testing. It should describe a standard scenario with inter-networking description (the interactions with other systems) without delving into everything that can go wrong. Another suggestion is that the requirement owner should write function test on at least external interfaces so that the requirement will be documented in a testable manner.

Feature status tracking: There should be a mechanism in the development process to document and communicate the decisions regarding consciously delayed features as it otherwise will create false positives in testing. Currently, the test team creates a defect report for missing functionality and it goes through an expensive process to handle them. Furthermore, there is no distinction between unintentionally and intentionally missing features.

Improved interaction between teams and cross-functional teams: System management is in between the customer and design/test teams and they have a good understanding of the customer needs. So they can be helpful to ask for missing information or explanation to avoid misinterpretation.

Similarly, a new initiative to involve ST teams early on in the process was welcomed by the practitioners. Now the testing team has a better understanding of features and can estimate the expected workload better. This early involvement has led to improved planning and reduced the time it takes the test team to understand the deliverables. If we have a cross-functional team then there will be opportunities to have early deliveries to test and have feedback sooner about problems and avoid expensive opinion/question defect reports.
4.6 Conclusions and discussion

We made four contributions in this study. First, we presented a study that focuses on testing challenges in SoS context. We mapped the challenges we found to three categories, 1) challenges that are not different in SoS context and other contexts, 2) challenges that are amplified in the SoS context but that can also be found in other contexts, and 3) to new challenges specific only to SoS context. To our knowledge, only Columbi et al. [12] has addressed this topic previously. In comparison to their SoS challenges we can say that their challenges (2) and (4) in Section 4.3 are not true in our case as the case company has tests to evaluate overall SoS capabilities and overarching test scenarios in compound system tests and field tests (see Figure 4.2). Their challenge (1) is partially true in our case as the current organizational structure does not support testing in SoS context, for example the lack of regression test suite maintenance can be traced back to the organization of our case company. Finally, their challenge (3) indicating that SoS systems have a high number of requirements leading to a high number of tests is completely true in our case. Reflecting their solutions proposals with the light of our case we can state that their proposal (2) using risk assessment to prioritize testing would be useful in our case as well. Their solution proposal (3) suggest focusing tests on interfaces might be applicable in our case but we think that other test focusing criteria should be used as well. Their solution proposal (1) “build little, test little” has already been applied and even exceeded in our case as the testing is following agile software development pace and testing with high coverage is performed continuously on different levels (see Figure 4.2). Their final proposal (4) to provide designated environments and people for integration does not seem that useful in our case as testing integrations did not seem to be a big problem and on the other hand designated people craved more support to regressions test maintenance rather than integration test.

Second, we found a contradiction between the software process improvement focus areas stemming from the fault slippage measure and interviews with practitioners. The fault slippage of internal processes and from customers clearly pointed to improving the areas of review and basic test that resulted in the highest fault slippage numbers (see Tables 4.5 and 4.6 ). However, the challenges and improvement ideas from practitioners pointed mostly to improving the functional tests and the system test suite that had poor maintainability and turnaround time, but high fault detection capability (see Figures 4.4 and 4.5 , and Table 4.7). On the other hand the reasons for problems in FT were caused at least partly by the fact that people did not do basic testing, but implemented the same test as functional tests because BT offered a poor technical support for testing. In the end, we cannot be certain if the company is better off improving functional or basic tests. We believe this can be generalized to the questions whether
companies are better off in improving the practices that are already strong, but still have good improvement potential, or the practices that are weak.

Third, we found an interesting circular relationship between the maintenance and turnaround time of functional test. It is difficult to improve turnaround time if the test code has low maintainability. However, the low maintainability also increases the turnaround time when developers add new test cases rather than modify the existing test cases to complement for new features. We believe that circular relationship can be attributed to the SoS context and the managerial independence that each development team has.

Fourth, we found that test case maintainability and maintenance are a big problem in SoS context. Future work should see how could the techniques and practices for regular software maintenance help test code maintenance in SoS context.
4.7 References


REFERENCES


Chapter 5

Aggregating software process simulation guidelines: Literature review and action research

5.1 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 modelling (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.
5.2 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 [23] [24]. The potential of SPSM is well established and it is time to make an engineering discipline from this art form [36]. 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. [36] for discrete event simulation (DES) and Pfahl and Lebsanft [25] for System Dynamics (SD). Yet other differences in the proposed processes are the level of formalism in the process description [36], use of terminology or targeted experience level of modeller [29] and size of the organization undertaking SPSM [33].

Some of these proposals suggest that their process is extensible to simulation approaches other than the one they were originally proposed for, such as [36]. It is not obvious to an inexperienced modeller 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 analyse 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 modelling in the software industry.

Ahmed et al. [31] found in a survey that software process simulation modellers consider the modelling 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 modellers do not report on the modelling process as either they already have a satisfactory process or are not interested in it. They [31] also found that most of the software process simulation modellers 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 learnt 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 synthesis of a modelling and simulation process from literature sources along with supplementary recommendations for each step of this consolidated process.
2. Secondly, this process, which utilizes experience of various simulators (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.

This chapter is an extension of a conference paper [22] that is extended in following respects:

- The literature review is supplemented by “forward referencing” of the primary studies to identify any work that builds upon or reports experience of using these guidelines. This ensured extended coverage of any more recent relevant literature.
- We have added the outcomes of the application of the consolidated process (with concrete output products from each step). This provides useful information for understanding the application of the process. Furthermore, useful references to each step of the process have been added from the SPSM and the simulation literature in general.
- A short description of the context describing the company, process and systems under development is also added.
- Section 5.4, now provides details of the research methods used in this study. A list of primary studies is also provided in the study for traceability.
- A detailed discussion of the validity threats and the limitations of the study are provided in Section 5.4.

The remainder of the chapter is outlined as follows. Section 5.3 summarizes the existing processes for SPSM and their industrial evaluation. Section 5.4 describes the research methods used in this study. Section 5.5 presents the aggregated guidelines
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

and consolidated simulation process. Section 5.6 reports the experience of using the consolidated process in the case company. Section 5.7 reflects on the findings of the literature review and the action research. Section 5.8 concludes the article.

5.3 Related work

Pfahl and Günther [26] proposed a methodology, Integrated Measurement, Modelling, and Simulation (IMMoS), which complements SD with existing modelling 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 sub-activities [25] [27]. Angkasaputra and Pfahl [38], analysed this process against agile practices and made improvements to make it more agile. Zhai et al. [32] 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. [36] 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 [34] summarize five major steps for SPSM studies and refer to [26] and [36] for detailed guidelines for SD and DES respectively.

Ahmed et al. [29] proposed a process based on interviews of simulation modellers. This process was aimed at novice modellers and it was evaluated in a controlled experiment with students as subjects.

Silva Filho et al. [33] 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. [24] 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 [23] 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. [13] 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); Modelling (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 modelling and DEVS-Hybrid for simulation of the model.

There is a large number of industrial SPSM studies however to the best of our knowledge, among the explicitly documented simulation processes only [26], [36] and [13] 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 artefacts for SPSM will be the basis to identify how simulation models can be developed for and with reuse [37] and will help reduce effort and cost of the SPSM.

5.4 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 modelling 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.
5.4.1 Literature review

The existing systematic literature review by Zhang et al. [45] [17] 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 [8] [7] we decided to supplement the search results of Zhang et al. [45] [17] studies by forward and backward snowballing. We used the Webster and Watson [20] three-step process in this study:

Step 1: Zhang et al. [45] [17] had reported a two phase systematic literature review scoping the research on the SPSM. They had reported [45] 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 [17] 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. [45] [17] review and identified articles relevant to the aims of our study.

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. [17] 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.

5.4.2 Data analysis

Fulltext of the articles identified above was read and process steps were extracted. The statements representing these steps were copied verbatim in a spreadsheet and analysed using the following three steps:

---

1 Since 2010 this journal has been incorporated in Journal of Software Maintenance and Evolution: Research and Practice.
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 catalogued.

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 consolidate process evolved based on the grouping. This reduced our influence on the formulation of the overall consolidated process.

5.4.3 Action research

We followed the action research methodology that is based on McKay and Marshall’s work [19] depicted as six phases marked as A, B, C, D, E and F in Figure 5.1. In the remainder of the section we provide details of each of the phases of the action research in this study.

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 [19]. 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 5.5.1.

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. The testing process and development context at the company are briefly discussed in Section 5.6.

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 5.5).

**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 [16]
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 [18]).

5.4.4 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 [15].

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 [15].

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 [20] 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 [8] [7].

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 [45] 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 [15].

Given that both authors had no previous experience of software process simulation modelling, 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 5.6) 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 5.7).
- All the contributing primary studies used to synthesise 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 [15].

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 5.7 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 [21].

Similarly, the current evaluation of the resulting model from use of this process can be only be considered as static validation [18]. 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.

5.5 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 [29] [34] [23]. The consolidation of simulation processes in literature resulted in a process with six steps and it is presented in Table 5.1 and Figure 5.2. We present a brief description of each step, the sources which recommend them, the model heuristics [23] and guidelines applicable for each of the steps (in Sections 5.5.1 to 5.5.6). We also report our experience from using this process in an industrial study in Section 5.6.

![Diagram of the consolidated process for conducting an SPSM study.](image)

Figure 5.2: Consolidated process for conducting an SPSM study.

Here are some of the guiding principles that are applicable in general to an SPSM:
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

- 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 [31] [30].
- Involve and maintain frequent contact with all stakeholders throughout the study [31] [30] [6].
- No model is perfect (i.e. it is an imitation and has a trade off between complexity and realism) [23].
- All models are incomplete (as the goal is to model the behaviour under study that is necessary to answer the questions of interest) [23].
- In industrial simulation studies it is important to deliver results and recommendations quickly as the modelled system and its environment are likely to change [30] [6].
- A model is not reality (therefore all results should be critically analysed) [23].
- 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) [23].
- Continually challenge the model (the credibility of the model can only be increased through further verification and validation V&V) [23] [30].
- The models are there to support managers in decision making, not to make the decisions for them [23].
- 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) [23].

5.5.1 Problem definition

The starting point for SPSM is the identification of a problem to be investigated [24] [34] [36] [29] [23] [27]. 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 [46]. The common purposes for SPSM are [24]: 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 [27] [29].
- 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
Table 5.1: Mapping of the steps in the consolidated process to the contributing literature.

<table>
<thead>
<tr>
<th>Process step</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem definition</strong></td>
<td>[24], [34], [36], [29], [23] and [27].</td>
</tr>
<tr>
<td><strong>Model design</strong></td>
<td></td>
</tr>
<tr>
<td>Model Scoping</td>
<td>[24], [29], [13] and [23].</td>
</tr>
<tr>
<td>Identifying input parameters</td>
<td>[24] and [36].</td>
</tr>
<tr>
<td>Identifying result variables</td>
<td>[24], [36], [13] and [29].</td>
</tr>
<tr>
<td>Specifying reference behaviour</td>
<td>[34] and [23].</td>
</tr>
<tr>
<td>Conceptual modelling</td>
<td>[24], [34], [36] and [13].</td>
</tr>
<tr>
<td>Choosing a simulation approach</td>
<td>[24].</td>
</tr>
<tr>
<td>Assessing technical feasibility</td>
<td>[27] and [29].</td>
</tr>
<tr>
<td><strong>Implementation of an executable model</strong></td>
<td>[34], [36], [13] and [29].</td>
</tr>
<tr>
<td><strong>Verification and validation</strong></td>
<td>[24], [36], [29], [34], [27] and [23].</td>
</tr>
<tr>
<td><strong>Simulation based investigations</strong></td>
<td>[34], [29] and [23].</td>
</tr>
<tr>
<td><strong>Presentation and documentation</strong></td>
<td>[29], [27] and [23].</td>
</tr>
</tbody>
</table>

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 [36].

- Develop test cases that clarify the model’s purpose and will be later used for validation of the simulation model itself and its results [36].

**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 [23].
- A model is created to answer specific questions [23], i.e. the SPSM undertaking should be goal-driven [25].
- 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 [25].
- A clear, well-specified purpose will guide the remaining process. It will also help avoid misuse of the model in the future [25] [23].
5.5.2 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 modelled are defined [24] [29] [23] [13]. 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 modelled. 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, modelling should proceed iteratively ensuring the delivery of the first executable model as soon as possible [25] [6].
- 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 [25].
- Use GQM for defining goals, questions and the appropriate metrics for simulation [36] [23] [13].

Identifying input parameters

Identification of the input parameters for the simulation model is proposed in [24] [36]. 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 [23]. Also note that it is often not possible to measure all variables of relevance accurately.
- Select input probability distributions for input variables [12]. Law [42] provides a detailed discussion of statistical methods to support this decision.
- Ensure integrity of data by maintaining constant communication with stakeholders [6].
Identifying result variables

Identify the outputs required from the model [24] [36] [29] [13] 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 [36] [23].

Specifying reference behaviour

The specification of reference behaviour is proposed in [34] [23]. Müller and Pfahl emphasize the importance of specifying observed problematic behaviour or the desired behaviour (called the reference behaviour) [34]. It can also help to identify the model output parameters and can be used later as input for model validation.

*Guidelines:*

- The reference behaviour should be defined (plotting behavioural change over time) [23] based on historical data [4].
- Consider behavioural patterns based on experience when actual data is not available in the company [23] [4].
- Instead of striving for absolute measures focus on relative measures (e.g. increase defect count in %) [23].

Conceptual modelling

Conceptual modelling is proposed in [24] [34] [36] [13]. In this step, process elements, information flows, decision rules and behaviour that have an influence on the result variables and are relevant to the simulation goal are identified for modelling. 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 artefacts in various activities [36].
- Create influence diagrams where (positive or negative) influence of various parameters on each other is depicted [36].
- Create causal loop diagrams (more pertinent for SD) [23] [27].
• Collect and analyse available empirical data for deriving the quantitative relations identified in the influence diagrams [36].
• Distinguish the parameter type as calibration, variable or project specific input parameter [36].

Guidelines:
• Consult domain experts as they have knowledge beyond the documentation and can judge the relevance of information to the problem under study [34].
• Interview domain experts during modelling to avoid missing important aspects of the process and reduce the threat of misunderstanding [25].
• Motivate the choice of cause-effect relations with data sources and literature [4].
• Do not try to model the “system”, i.e. include only the entities and relations necessary to generate the behaviour of interest. Using a top-down iterative approach, add details only when necessary [23]. Start small and later on add more content to the simulation model [31].
• Using the goal and the scope of model, aggregate and abstract to an appropriate degree hiding unnecessary details [23].
• Keep both the model and the communication with users simple [23]. Review each individual cause-effect relation in the causal diagram before the combinations [4].

Choosing a simulation approach

Choosing a simulation approach is proposed in proposed in [24]. 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 [23]. The choice of an appropriate simulation approach will therefore depend on the aim of the study and the modelled system. A multitude of simulation approaches are available. Zhang et al. [45] 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 analyses and long-term behaviour. SD models have levels and flows of entities. These entities are not individually identified and are not traced through the process [24]. 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 [24].
Assessing technical feasibility

Establishing technical feasibility is proposed in [27] [29] i.e. prerequisites for simulation model development and usage like adequacy of problem definition, availability of data, and process maturity [4]. Before the simulation model development technical feasibility should be checked [27] [29]. Ahmed et al. [31] found that most of the modellers assess feasibility as a first in the modelling process.

Guidelines: The feasibility aspects identified in [4] are covered by the following questions proposed by Balci [12]:

- 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?

5.5.3 Implementation of an executable model

Implementation of an executable model is the next step in the process as proposed in [34] [36] [13] [29]. 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. [36] 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. [30] in a survey also found that most simulation modellers 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 [23] [31] [30].
- The use of relative measures and normalizing them helps in scaling the model. [23].
- Develop the simulation model with high modularity [31]. Separate data from model to support modification and experimentation [6].
- Always keep the model in a state that it could be simulated (or tested) [23].

Choosing simulation tools and techniques

Another important decisions is the choice of simulation tool as proposed in [24] [29]. 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 [23] and Ahmed et al. [30] 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 [23] [24]. To facilitate data collection simulation model may be integrated with the domain systems [6]. However, a major issue than integration is the lack of data for model calibration and validation in real-world settings [24]. The desired data is often poorly defined, inaccurate or missing altogether.

**Guidelines:** To deal with a lack of project data Park et al. [13] recommend use of analytical models like COCOMO II [9].

Here are some suggestions to deal with the typical situations that modellers have to face:

- Consult domain experts to deduce the accuracy and relevance of data [6] [46].
- 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 [46].
- 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 [46].

### 5.5.4 Verification and validation

Verification and validation of the simulation model had been proposed in [24] [36] [29] [34] [27] [23]. A simulation model should undergo validation to the extent possible [24] and V&V should go on throughout the simulation study [36] [30]. 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 [46]. Before the full-scale development of the simulation model the following steps should be done:
• Validation of the SPSM requirements [36] identified problem against the formulated problem [12].
• Review of causal diagrams with domain experts [27].
• 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 [23]. Use presentations, reviews and meetings with the stakeholders to communicate any change in the model or data [6].
• 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 [23].

Guidelines: Seven V&V activities are presented in [46]. Ahmed et al. [28] 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 [42], [41] [2] [12]. Some guidelines from other researchers are:

• Validation should be ongoing as parts of model are developed [30].
• Assure face validity by using inspections and structured walk-through of the model [24] [36] [29].
• False expectations (e.g. perfect predictions on the first model run) should be avoided, rather patterns should be reviewed for qualitative similarity instead [23].
• 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 behaviour [23].
• Comparing simulation output to system output (actual data) [29]. See [42] for statistical methods to perform this comparison.
• Model validity is a relative matter [23] i.e. depending on the purpose of the study.

Sensitivity analysis
Sensitivity Analysis has been suggested as a V&V method [24] [29] [23]. It is also used as a method to design scenarios to investigate the behaviour of modelled system with extreme values. It is applicable to all types of simulation approaches [24]. 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 [46].

Guidelines:
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

- Madachy [23] discusses application of numerical, behavioural and policy sensitivity analysis on simulation models.
- Begin with changing values of one parameter at a time keeping others constant [23].
- Statistical analysis is facilitated by good interconnectivity (automated data transfer) between the simulation and statistics tool [35].
- For stochastic models use Monte-Carlo analysis [23].

5.5.5 Simulation based investigations

Simulation based experimentation is proposed in [34] [29] [23]. 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 [46] explored various analytical and statistical techniques to evaluate process alternatives using simulation results. They [46] also showed how techniques like Data Envelopment Analysis (DEA) [5] can be used to perform impact analysis of such decisions.

Law [42] provides a good discussion of statistical experiment design and confidence interval based analysis of alternatives.

5.5.6 Presentation and documentation

Presentation of the simulation results and documentation of the study and the model is proposed in [29] [27] [23]. 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 [12]. 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 [23] and from scientific publishing perspective, please refer to França and Travassos reporting guidelines [10].

**Guidelines:**

- Describing the goal and scope of the model [23] [10].
- Documenting causal relations and feedback loops [10]. Like any software, well-commented equations are easier to understand [23].
- Document assumptions, calibration values and rationales [10] [23].
- Verification and validation procedures and their outcome [10].
- Show output of multiple scenarios and discuss them [23]. The relevance and adequacy of these scenarios should also be motivated [10].
- Document any constraints and limitations in implementing a conceptual model in a simulation package [10].
- Like other empirical research, simulation studies should discuss the threats to study validity [10].

### 5.6 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. Furthermore this description of the test process will help understand the simulation model that was developed in this study (presented in Section 5.6.2).

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 5.3. 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 5.3). 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 [21] the following should be added as information:

- First versions of all systems are older than 5 years.
• 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 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.

5.6.1 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 5.2.

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 modelling 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).

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 GQM 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 modellers correctly interpreted the goals and questions that the model should be able to answer.

Table 5.2: Goal of the simulation study at the case company.

<table>
<thead>
<tr>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>To develop a better understanding in developers of the impact of missing system-test iteration on the compound-test and the overall system.</td>
</tr>
<tr>
<td>- Demonstrate the impact of a requirement or a defect-report missing the system-test or compound-system test?</td>
</tr>
<tr>
<td>- Demonstrate the impact of a requirement with dependencies missing the system-test or compound-system test?</td>
</tr>
<tr>
<td>- Demonstrate the benefits of early integration?</td>
</tr>
</tbody>
</table>

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).

5.6.2 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 modelled it was important to see what level of dependence exists between the subsystem being modelled 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 5.6 and shown in Figure 5.3. 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 5.6.2.

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 5.3, 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 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 5.5.1) 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 behaviour

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 behaviour of the testing process. Apart from the benefits of reference behaviour for validation and output parameter identification, we found that being able to replicate the reference behaviour had a very positive impact
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

Table 5.3: Data collection.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable name</td>
<td>Name the variable e.g. Size</td>
</tr>
<tr>
<td>Description</td>
<td>A short description of the variable.</td>
</tr>
<tr>
<td>Variable type</td>
<td>Depending on how we want to analyse this variable as independent, dependent or control variable.</td>
</tr>
<tr>
<td>Stakeholders (roles)</td>
<td>The stakeholders responsible for this variable or who could give more information about this variable.</td>
</tr>
<tr>
<td>Activities</td>
<td>The activities in the software development lifecycle that will influence, produce or use this variable’s values.</td>
</tr>
<tr>
<td>Software artefacts</td>
<td>Software artefacts where this variable can be computed from or is stored in.</td>
</tr>
<tr>
<td>Unit of measurement</td>
<td>The unit of measurement.</td>
</tr>
<tr>
<td>Simulation values</td>
<td>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.</td>
</tr>
<tr>
<td>Typical and extreme values</td>
<td>The typical, minimum and maximum value of this variable.</td>
</tr>
<tr>
<td>Distribution</td>
<td>The distribution of this variable.</td>
</tr>
<tr>
<td>Tolerance</td>
<td>The acceptable tolerance in the estimation, in case of a dependent variable.</td>
</tr>
</tbody>
</table>

on the interest levels of simulation-model users. As in our case the users were excited to see that the system behaviour 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 modelling

What cause/effect relations should be modelled? 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 modelled 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 analysed the empirical data early on in the study rather than waiting till the process modelling was done [36]. 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 5.4.

![Diagram of the testing process](image)

**Figure 5.4: Conceptual view of the testing process at a high level.**
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

Choosing a simulation approach

The decision of the simulation modelling 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 artefacts 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 modelling both the discrete and continuous aspects of the process. In addition to the aforementioned reasons for starting with SD modelling, 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 modellers we did not feel confident to comment on the “technical feasibility” before the initial model development as recommended by [27]. 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 5.6.3 and we were able to develop the testing process simulation model.

5.6.3 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 [45]. 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 5.5.3 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 5.3 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 behaviour 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.

5.6.4 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 [36] 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 modelling the process and reasons for making them. The assumptions and simplification in this simulation model are summarised in Table 5.4.

We found developing concrete questions and usage scenarios was very helpful to validate that we (as modellers) 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 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.

Table 5.4: 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 modelled. |
| The prioritization from the management for development will be modelled 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. |

We followed the presentation with a structured interview to assess the validity of the simulation model with practitioners having different roles in the organization and hence brought different perspectives of the software test process. The questions used in the interviews are listed in Table 5.5. We used face validity of graphical models, replication of trends in reference behaviour, 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 behaviour 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
Chapter 5. Aggregating software process simulation guidelines: Literature review and action research

sense in all three cases or not.

Table 5.5: Questions used in the validation of the simulation model.

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you think the aim has been correctly interpreted?</td>
</tr>
<tr>
<td>Are the criteria (output and scenarios) to evaluate the aim of the study complete?</td>
</tr>
<tr>
<td>Does the model correctly represent the current testing process at the company e.g. no missing phase or inconsistent terminology?</td>
</tr>
<tr>
<td>Is the model missing a cause/effect relation or a feedback loop?</td>
</tr>
<tr>
<td>Are the simplifying assumptions reasonable and acceptable from their perspective?</td>
</tr>
</tbody>
</table>

5.6.5 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 5.5. 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

168
(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 5.5: Output of the simulated scenarios using different variables

The effects of these failure rates are visualised on workload in Figure 5.5a, productivity in Figure 5.5b and delivery to customer in Figure 5.5c. 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 5.5b) and in this case the company
does not even deliver to its full capacity (see the blue curve in Figure 5.5c).

This model will be used to demonstrate that early integration results in improved
flow of artefacts and consistent delivery to the customer.

5.6.6 Presentation and documentation

We used Madachy’s [23] report template as a checklist when documenting the simula-
tion 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 abstrac-
tion of reality, the results were analysed critically and other competing explanations for
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.

5.7 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. [36]). Most of the methodologies found in this study are incremental and
iterative in nature, (see e.g. [29] [34] [23] [26]).

We have found that the overall process for SPSM is very similar and independent
of the different simulation approaches, experience level of modellers, and the size of
the organizations’ where the study is conducted. By consolidating the individual pro-
cess 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., [13] is useful for SPEM based simulation. The con-
solidated process in such cases contributes by highlighting what is still important e.g.
V&V in case of [13] 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 [43] in
this forum (as shown in the second column of Table 5.6) to the SPSM-literature based
consolidated process description (as shown in the first column of Table 5.6). 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 5.6) we may conclude that the overall simulation process is independent of the organizational context, simulation approach and experience of the modellers. 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.

<table>
<thead>
<tr>
<th>Steps in consolidated process for SPSM</th>
<th>Shannon’s simulation process [43]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Problem definition</td>
<td>1. Problem definition</td>
</tr>
<tr>
<td>B. Model design</td>
<td>2. Project planning</td>
</tr>
<tr>
<td>1. Model scoping</td>
<td>3. System definition and determining the boundaries</td>
</tr>
<tr>
<td>2. Identifying the input parameters</td>
<td>4. Conceptual model formulation</td>
</tr>
<tr>
<td>3. Identifying the result variables</td>
<td>5. Preliminary experimental design</td>
</tr>
<tr>
<td>6. Choosing a simulation approach</td>
<td>8. Verification and Validation</td>
</tr>
<tr>
<td>C. Implementation of an executable model</td>
<td>10. Experimentation</td>
</tr>
<tr>
<td>1. Choosing the simulation tools and techniques</td>
<td>11. Analysis and interpretation</td>
</tr>
<tr>
<td>2. Model calibration</td>
<td>12. Implementation and documentation</td>
</tr>
<tr>
<td>D. Verification and validation</td>
<td></td>
</tr>
<tr>
<td>E. Simulation based investigations</td>
<td></td>
</tr>
<tr>
<td>F. Presentation and documentation</td>
<td></td>
</tr>
</tbody>
</table>

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:

- The work on V&V of simulation models by Balci [2] is useful. Similarly, Withers [3] has developed a mapping of best practices from software engineering to the practice of DES.
- The guidelines and tips given by Robinson [44] for the conceptual modelling activity, a vital part of a simulation study, are also valid and applicable for SPSM and have some overlap with Madachy’s “Modelling heuristics” [23]. Balci [12] 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. [14] 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 [11] 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. [14] 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 consolidate process presented in this article) 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.

### 5.8 Conclusion

This article 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 [36] [29] 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 modellers 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 ([43] and [12]) 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).
5.9 References


REFERENCES


REFERENCES


ABSTRACT

Background: Since the 1950s explicit software process models have been used for planning, executing and controlling software development activities. To overcome the limitation of static models at capturing the inherent dynamism in software development, Software Process Simulation Modelling (SPSM) was introduced in the late 1970s. SPSM has been used to address various challenges, e.g., estimation, planning and process assessment. The simulation models developed over the years have varied in their scope, purpose, approach and the application domain. However, there is a need to aggregate the evidence regarding the usefulness of SPSM for achieving its intended purposes.

Objective: This thesis aims to facilitate adoption of SPSM in industrial practice by exploring two directions. Firstly it aims to establish the usefulness of SPSM for its intended purposes, e.g., for planning, training and as an alternative to study the real world software (industrial and open source) development. Secondly to define and evaluate a process for conducting SPSM studies in industry.

Method: A literature review, two systematic literature reviews (SLR), a case study and an action research study were conducted. A literature review of existing SLRs was done to identify the strategies for selecting studies. The resulting process for study selection was utilized in an SLR to capture and aggregate evidence regarding the usefulness of SPSM. Another SLR was used to identify existing process descriptions of how to conduct an SPSM study. The consolidated process and associated guidelines identified in this review were used in an action research study to develop a simulation model of the testing process in a large telecommunication vendor. The action research was preceded by a case study to understand the testing process at the company.

Results: A study selection process based on the strategies identified from literature was proposed. It was found to systemize selection and support inclusiveness with reasonable additional effort in an SLR of the SPSM literature. The SPSM studies identified in literature scored poorly on the rigor and relevance criteria and lacked evaluation of SPSM for the intended purposes. Lastly, based on literature, a six-step process to conduct an SPSM study was used to develop a system dynamics model of the testing process for training purposes in the company.

Conclusion: The findings identify two potential directions for facilitating SPSM adoption. First, by learning from other disciplines having done simulation for a longer time. It was evident how similar the consolidated process for conducting an SPSM study was to the process used in simulation in general. Second the existing work on SPSM can at best be classified as strong “proof-of-concept” that SPSM can be useful in the real world software development. Thus, there is a need to evaluate and report the usefulness of SPSM for the intended purposes with scientific rigor.