

Exploring Processes in Small Software Companies: A Systematic Review

Nirnaya Tripathi^(✉), Elina Annanperä, Markku Oivo^(✉),
and Kari Liukkunen

M3S Research Unit, University of Oulu, Oulu, Finland
{Nirnaya.Tripathi, Elina.Annanpera,
Markku.Oivo, Kari.Liukkunen}@oulu.fi

Abstract. Context: To attain an advantage over competitors, small software companies (SSCs) need to have an efficient software development process. However, systematic review studies that have examined the software development process within the context of SSCs are limited. Objective: Therefore, the objective of this paper is to rigorously assess the current state of practice of the software development process of SSCs using ISO/IEC 12207 standard as an analyzing framework. Method: A systematic literature review was conducted to analyze relevant papers published between 2004 and 2014. The selected papers were categorized according to the empirical technique used. A total of 41 primary papers focusing on various aspects of the software development process of SSCs were discovered out of 3841 papers. Results: Based on the evidence found in primary papers, requirement engineering, project planning, life cycle model management and configuration management are the frequently considered processes for improvement when software process improvement (SPI) programs are conducted in SSCs. In addition, understanding the collected requirements and communication barriers between product management and the rest of the work team were among the challenges observed and experienced by SSCs during the software development process.

Keywords: Software process · Small software company · Systematic literature review · ISO/IEC 12207

1 Introduction

In the global economy, the most rapidly growing sector is the software industry, which has emerged as one of the key economic drivers for many nations [1]. Another significant entity that contributes to driving the world economy is the small software company (SSC). SSCs play an important role in their nations' economies because of their ability to capture the kinds of markets that larger companies are incapable of reaching or have rejected. SSCs typically consist of less than 50 employees, and their aim is to create one or a few software products for their customers [2]. They also develop components for larger systems produced by other companies or offer maintenance services for the software products created by other firms [3].

To maintain their competitiveness in their target markets and to sustain a healthy relationship with their customers, SSCs need to supply increasingly faster and cheaper software products [4]. To develop such software products, these companies need to have efficient software development processes [5]. SSCs definitely need all the assistance in seeking the relevant information to make their software development processes efficient, but they often lack systematic process knowledge for determining which type of processes are more relevant to their context [3, 6]. One way to analyze software process knowledge is by systematically examining the existing literature on SSCs. Some systematic literature reviews (SLRs) have focused on analyzing the existing approaches toward software process improvement (SPI) in small and medium enterprises (SMEs). For example, Pino et al. [7] concluded that models such as Capability Maturity Model Integration (CMMI) and Software Process Improvement and Capability Determination (SPICE) are not suitable for SPI programs in SMEs and instead proposed the use of lighter-weight models. Sulayman and Mendes [8] concluded that very few studies have explicitly focused on SPI for Web companies. Paternoster et al. [9] and Klotins et al. [10] described the software development of start-ups. However, systematic review studies in the context of SSCs and their software development processes are limited [11]. Therefore, there is a need for systematic review studies in the SSC context with the goal to explore the state-of-practice of software processes and the types of challenges associated with it.

In this paper, we present a systematic review of the literature, with the aim of identifying the software processes utilized in SSCs. We have used the ISO/IEC 12207 [12] processes as an analyzing framework to describe both the current state-of-practice and challenges present in the software development processes. The paper proceeds as follows. In Sect. 2, the background and motivation are described. In Sect. 3, the systematic review is presented. Section 4 presents the results from the review, followed by a discussion of the research questions presented and our conclusions.

2 Background and Motivation

SSCs are considered as an important entity in the software industry and they represent up to 85 % of all software companies [13, 2]. They often face vicious competition from their competitors in developing quality software products on a strict deadline to fulfill customer requirements [2]. Their software development activity is generally in casual state [14]. Therefore, in order to improve their productivity, they need to improve their software development processes [15]. However, SSCs typically avoid process improvements [15]. This avoidance mainly appears due to the lack of financial and human resources [14]. To help SSCs, it is important for the software engineering research community to explore software processes in detail. Doing this will provide added value to the practitioners operating in the small-scale context.

There are some SLRs that have focused on the improvement of software development processes in SMEs. For example, Pino et al. [7] analyzed the existing approaches toward SPI in SMEs. They concluded that proper SPI programs based on models such as CMMI and SPICE are not suitable for SMEs and proposed the use of lighter-weight models. They also observed in their study that frequently improved

processes in SPI programs are project management, documentation, requirement change management, process establishment, configuration management, and requirement elicitation. Sulayman and Mendes [8] conducted a systematic review study to determine the current state of research in SPI models and techniques used by small and medium Web development companies. They found that very few studies have explicitly focused on SPI for Web companies despite the large number of Web companies across the globe. Paternoster et al. [9] conducted a mapping study to explore the state of art of software development in start-ups. Their conclusion was that software engineering work practices are chosen based on the start-up context. Klotins et al. [10] also conducted a mapping study, in which they focused on the software engineering aspect of start-ups. They found that very few research papers have provided concrete evidence of software engineering knowledge areas in start-up companies.

However, systematic review studies in the particular field of SSCs focusing on the software development processes are limited [11]. To the best of our knowledge, an in-depth analysis of the studies reported in this context that describes the state-of-practice and the associated challenges with it does not appear to exist. To address this gap in the literature, we therefore decided to conduct a systematic review of the literature that focuses on software development processes in SSCs to explore this topic area. To analyze the software development processes in SSCs within a standardized framework, we decided to follow the classification of processes as defined in ISO/IEC 12207 [12]. ISO/IEC 12207 includes a process reference model that categorizes the processes related to software system into seven groups: agreement processes, organizational project-enabling processes, project processes, technical processes, software implementation processes, software support processes, and software reuse processes [12]. We did not consider ISO/IEC 29110 [16] as our analyzing framework since the standard is applicable to very small entities (up to 25 people), whereas our study focus is on SSCs (up to 50 people).

3 Research Methodology

Our literature review is based on the guidelines provided by Kitchenham and Charters [17]. We also used a software tool (StArt) to support our systematic review [18].

3.1 Research Question

The main goal of our review is to find all the research studies that relate to the software development processes of SSCs and that describe the key processes used and any related challenges. To achieve this goal, the following research questions (RQs) are defined:

RQ1: What is the state-of-practice in terms of the use of processes in small software companies? The outcome of this question is the current state of the processes used in SSCs. The state-of-practice is analyzed using the ISO 12207:2008 life-cycle processes.

RQ2: *What are the challenges that small software companies face in the processes?*

The outcome of this question is the discovery of the challenges associated with the software development processes in SSCs.

3.2 Search Strategy and Data Retrieval

The initial sets of keywords for SSCs were taken from the work of Pino et al. [7]. Pilot searches in the key software engineering research domain databases were conducted, and new keywords were added to better target the searches to the desired data set. In the end, the search strings were formulated by combining the terms representing the population AND intervention. It can be summarized as follows: $(X1 \text{ OR } X2 \dots \text{OR } Xn) \text{ AND } (Y1 \text{ OR } Y2 \dots \text{OR } Yn)$, where X represents the population and Y represent intervention. The population (X) is represented as {software engineering process, software process, software engineering}, whereas intervention (Y) is represented as {small company, small enterprises, small organizations, small team, and small settings}. During the data retrieval process, the publication period was set to 2004 to 2014, with the purpose of summarizing the most recent related work. We used a total of four databases in our study:

- Scopus (<http://www.scopus.com/>)
- IEEE Xplore (<http://ieeexplore.ieee.org>)
- Science Direct (<http://www.sciencedirect.com/>)
- Web of Science (<http://www.webofknowledge.com>).

3.3 Study Selection and Data Extraction

The database searches resulted in identifying 7967 papers, out of which 4022 were duplicates. The remaining 3945 papers were screened and assessed by two researchers. The inclusion/exclusion criteria were determined based on title, abstract, and keywords, implicitly or explicitly (see Table 1). The papers were categorized as “accepted”, “rejected” or “can’t decide” (in StArt terms, “unclassified” is used to the same extent). The “can’t decide” papers were discussed, and issues were resolved between the first and second author. After that, 104 papers were selected for full text reading. Thus, 41 studies were finally selected as the primary study papers. For quality assessment, we used the systematic and validated model [19] to assess the scientific rigor and industrial relevance of each primary study. For scientific rigor, we considered the following aspects: context, study design, and validity. For industrial relevance, we considered the following aspects: subjects, context, scale, and research method. More details about the papers, based on scientific rigor and industrial relevance, are shown in Fig. 1. Data extraction and analysis was done through qualitative data analysis using the NVivo software. We used the deductive approach for coding the primary studies [20].

Table 1. Inclusion/exclusion criteria

Inclusion criteria	Exclusion criteria	Articles
(1) Reports an empirical study and/or (2) reports a study on small software companies and the software development process	Title, Abstract, Keywords: (1) The main focus was not on the small software companies and/or the software development process; or (2) the paper was not available; or (3) it was not in English; or (4) the paper was a letter, editorial, or position paper	104
	Full Text Read: (1) No reported result or (2) same content as other studies (extended papers, summaries)	41

3.4 Validity Discussion

In this section, we discuss the validity of our study in terms of construct validity, internal validity, external validity, and reliability [21]. Construct validity deals with whether the primary study papers focused on the software development processes and SSCs. During the search process, search strings were designed in such a way that they would collect a wide variety of papers related to this topic. To collect the papers, databases that are frequently used for accessing software engineering literature were used. During the full-text selection phase, manual screening was conducted twice to ensure that only papers relevant to the research questions were included. Reliability concerns whether the study can be repeated by the other researchers. Our study was based on a systematic review process protocol with well-defined search strings in the most common databases through the use of the SLR tool. Therefore, our review process can easily be reproduced by other researchers. The inclusion and exclusion criteria were piloted and followed during the review process. Internal validity refers to problems in the analysis of the data. During the execution phase, some primary study candidate papers did not show clear objectives and results, which made inclusion/exclusion criteria difficult. To overcome this situation, several meetings were arranged between the authors to evaluate the discrepant papers and eliminate the ones that did not match the inclusion/exclusion criteria. External validity concerns whether the results of an SLR can be generalized. In this situation, the results from this study were limited to the SSC context; therefore, both researchers and industrial communities interested in the software development processes of SSCs can benefit from our findings.

4 Results

As a result of the systematic review, a total of 41 primary studies were found to be relevant to the RQs. The descriptions of the 41 papers are shown in Table 2. In the following section, we give an overview of the primary studies. We then explore the

state-of-practice and challenges associated with SSCs' processes. To investigate the software development processes of SSCs in a systematic way, we classified the processes as defined in the ISO/IEC 12207 [12].

Table 2. List of primary studies

ID	Source	Year	Method	ID	Source	Year	Method
[3]	Scopus	2006	Case study (CS)	[22]	Scopus	2012	Ethnography
[4]	Scopus	2008	Case study	[23]	Scopus	2007	Case study
[6]	IEEE	2012	Case study	[24]	Scopus	2014	Case study
[25]	Scopus	2008	Case study	[26]	Scopus	2010	Case study
[27]	Scopus	2010	Case study	[28]	Scopus	2007	Experiment
[29]	Scopus	2010	Case study	[30]	IEEE	2012	Experiment
[13]	Scopus	2013	Action research	[31]	Scopus	2006	Case study
[32]	Scopus	2006	Case study	[33]	Scopus	2010	Survey
[14]	Scopus	2010	CS/Survey	[34]	Scopus	2009	Case study
[15]	IEEE	2009	Experiment	[35]	IEEE	2008	Case study
[36]	Web of science	2008	Survey	[37]	Web of science	2008	Experiment
[38]	Scopus	2013	Action research	[39]	Scopus	2005	Case study
[40]	WOS	2008	Case study	[41]	Scopus	2004	Survey
[42]	Scopus	2013	Case study	[43]	Scopus	2004	Case study
[44]	Scopus	2005	Case study	[45]	Scopus	2009	Case study
[46]	Scopus	2013	Survey	[47]	Scopus	2011	Case study
[48]	Scopus	2012	Case study	[49]	Scopus	2010	Case study
[50]	IEEE	2010	Case study	[51]	IEEE	2007	Case study
[52]	WOS	2004	Survey	[53]	Scopus	2012	Case study
[54]	IEEE	2006	Case study	[55]	Scopus	2011	Survey
[56]	Scopus	2009	Experiment				

4.1 Overview of the Studies

Rigor and Relevance. The quality of the primary studies was evaluated according to rigor and relevance, as discussed in Sect. 3.3. As can be seen in Fig. 1, thirteen studies are in the upper-right quadrant (Rigor: High, Medium and Relevance: High, Medium) in the most appropriate region. Sixteen studies exhibited high industry relevance (Relevance: High), out of which ten showed low scientific rigor. Twenty-five studies exhibited moderate industry relevance, out of which seven demonstrated medium rigor and eighteen demonstrated low rigor. Based on this evaluation, we established the fact that most of the identified studies were conducted in cooperation with an actual SSC, thus scoring medium or high on the relevance scale. However, many of these studies had low rigor.

Research Methodology. Most of the studies focused on empirical evaluations of theoretical concepts related to the software development processes and were applied in an SSC context, with the overall goal of assessing and/or improving software development processes. In Fig. 1, it can be seen that the majority of the primary studies are listed in the case study category. The other common methods were surveys and experiments. Also, looking at the left-lower quadrant in Fig. 1, it can be seen that the majority of the papers (28; 18 case studies, 5 surveys, 4 experiments, and 1 ethnographic study) fell in the low rigor category. Therefore, generalizability of the results of the given studies is low.

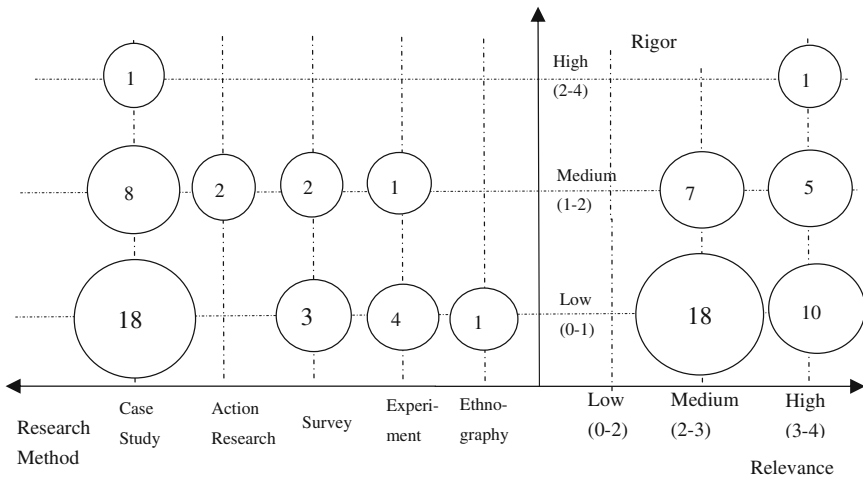


Fig. 1. Overview of research method, rigor, and relevance distribution of papers

4.2 Processes

A detailed overview of processes as defined in the ISO/IEC 12207 and relevant primary study papers are shown in Table 3.

SW Implementation Processes. Processes related to requirements are often considered critical processes for improvement since SSCs aim to quickly deliver what their customers want using their own practices rather than including customer commands in their development process [15]. Generally, functional requirements are the main focus during the requirement gathering of the software product [52]. One challenge that SSCs experience is understanding what the collected requirements really mean. The gathered information needs to be clearly understood to change it into clear product requirements. Clear understanding is crucial because it is important to determine which requirements match the business goal. Without a clear understanding, it is difficult to determine which product requirements need to put into which release plan to meet the business goals. [27] In respect to software qualification testing, some studies have reported that some companies have had troubles with testing their products [24, 25]. One problem

was related to a deficiency of testing knowledge in the company. This lack of knowledge results in a lack of test plans, which causes uncertainty in companies during the testing process [25]. This lack of knowledge can be due to the absence of defined process descriptions or the quality of instructional manuals. However, Clarke and O'Connor [46] found that a majority of companies report general improvements in testing, including a periodic increase in test suites, the creation of a standardized testing process, and more emphasis on performance testing. Another improvement reported was measuring more code coverage of the automated tests [46].

SW Support Processes. In regard to the software quality assurance (QA) process, Wilkie et al. [44] surveyed six companies and found that they each defined QA differently. For some, QA means testing, while for others, it is an instrument to assess a software product [44]. This confusion may be due to a lack of proper process knowledge. Regarding software configuration management (CM), one study [52] found that SSCs are able to perform basic CM tasks (i.e., version control, change management, and release management). One strength of the CM process in most of the companies was the code version control [52]. Some challenges with CM were also reported, such as uncertainty surrounding which work products should be with what version control and the lack of appropriate guidelines for CM [52].

Table 3. Processes and relevant papers

Life cycle processes	Primary study and frequency percentage	
Software specific processes		
<i>SW implementation processes</i>		
Software implementation	–	
Software requirements analysis	[4, 15, 22–24, 26, 31, 32, 37, 39–46, 51, 56, 27]	49 %
Software architectural design	[24]	2 %
Software detailed design	[32, 43, 56]	7 %
Software construction	[3, 24, 37, 38, 41, 43, 56]	17 %
Software integration	[43]	2 %
Software qualification testing	[23–25, 31, 37, 38, 43, 46, 56]	22 %
<i>SW support processes</i>		
Software documentation management	[23, 25, 49]	7 %
Software configuration management	[22, 24, 26, 41, 43, 44, 49, 51, 52]	22 %
Software quality assurance	[22, 24, 26, 39, 41, 43, 44]	17 %
Software verification, validation	–	

(Continued)

Table 3. (Continued)

Life cycle processes	Primary study and frequency percentage	
Software specific processes		
Software review, audit	–	
Software problem resolution	[41, 43, 3, 15, 31, 46, 49]	17 %
Software reuse processes	–	
Domain engineering	–	
Reuse asset management	–	
Reuse program management	–	
System context processes		
Agreement processes		
Acquisition process	–	
Supply process	[3, 31, 46, 27]	10 %
Organizational project-enabling processes		
Life cycle model management	[11, 27, 37, 48, 49, 4, 6, 14, 16, 24, 47, 50, 51, 55, 22, 25, 32–34, 39, 42, 45, 53]	56 %
Infrastructure management	[46, 49]	5 %
Project portfolio management	–	
Human resource management	[37, 49, 55]	7 %
Quality management	[37, 52]	5 %
Project processes		
Project planning	[6, 15, 22–24, 26, 37, 39, 40, 43–46, 50, 51, 54]	39 %
Project assessment and control	[15, 24, 26, 39, 40, 44, 45, 51]	20 %
Decision management	–	
Risk management	[41, 43]	5 %
Configuration management	–	
Information management	–	
Measurement	[26, 34, 39, 44, 53]	12 %
Technical processes		
Stakeholder requirements definition	–	
System requirements analysis	[43]	2 %
System architectural design	[50]	2 %
Implementation	–	

(Continued)

Table 3. (Continued)

Life cycle processes	Primary study and frequency percentage	
Software specific processes		
System integration	[43]	2 %
System qualification testing	[43]	2 %
Software installation	[3, 31, 46, 49]	10 %
Software acceptance support	–	
Software operation	[3, 30, 31, 43]	10 %
Software maintenance	[24, 30, 34, 43, 46]	12 %
Software disposal	[46]	2 %

Agreement Processes. In the case of SSCs, no study has explicitly reported the use of processes within the acquisition process. The reason could be that SSCs are usually the ones who initially approach the customer. Since the amount of customers of SSCs is limited, this aspect is not well defined or of much interest. Regarding the supply process, Wangeheim et al. [3] found in their case study that this was one of the priority processes for improvement. In their other study [31] with eight companies, they found out that some of the companies had established systematic supply processes and had developed contract templates to improve relationships with their customers. Clarke and O'Connor [46], in their survey of 15 SMEs, found that those companies had made extensive improvements, specifically in the area of tendering, as well as in negotiating with customers.

Project Processes. On the subject of project processes, two studies [3, 32] that assessed SSCs reported that project management is considered a high-priority process. Saastamoinen and Tukiainen [52] pointed out that project planning is an important aspect in SSCs. In addition, within the project plan, the development approach should be clearly described. For SSCs, a project is managed in an iterative or incremental way, tending to follow the waterfall model. A project's work capacity and time approximation are usually calculated based on the project manager's experience.

Organizational Project-Enabling Processes. In relation to process establishment within life cycle model management, O'Connor et al. [29] explained that process establishment in SSCs is based on two categories: software development manager work background and market requirements [29]. From the perspective of an SSC, the benefits provided by a more standardized process could include increased competitiveness, superior customer satisfaction, and greater product quality [36]. Regarding process assessments, one study [14] reported that SSCs usually avoid the adoption of any process standards in their software development process. This may be due to the opinion that process standards are overly complex. However, in the context of process improvement, the objective for SSCs is to improve efficiency within the organization, improve productivity, and reduce development time [43]. SSCs tend to have problems with effectively improving their software processes [13], and the major difficulty could be a struggle to change [54]. In the context of knowledge management (KMP) within

the human resource management process, Basri and O'Connor [55] stated that SSCs have clear KMPs within their organizations. They pointed out that knowledge waste is not a serious problem in a small-scale context. Also, informal communication and macro management can assist in creating team dynamics, which can further heighten KMP [55, 49]. Concerning quality management, requirements for ensuring that SSCs meet a certain level of quality are lacking. To improve the quality of their products, a clear definition of quality is needed, and goals for quality management need to be set. These quality goals can be altered based on the project [52]. In the context of software maintenance within technical processes, one study observed that software maintenance is poorly implemented in some companies and needs to be a priority for further improvement [24]. A number of companies have reported that they have amplified the refactoring level to reduce future support and maintenance costs [46].

Challenges in Process. Some challenges were also observed in the primary study papers. A list of the challenges observed and experienced by SSCs within the processes is presented in Table 4. The challenges are mostly related to the following processes: software requirements analysis, software qualification testing, software quality assurance, software documentation management, software configuration management, and life cycle model management.

Table 4. Challenges in processes

Process	Challenge	Description
Software requirements analysis	Understanding gathered information	The gathered information needs to be clearly understood to determine its business value and to further develop it into an explicit product requirements for release plan [27]
	Communication gaps	This is difficulty in sharing information between the customers, product management team, and the rest of the organization [27]
	Nonfunctional requirements	Nonfunctional requirements such as performance and the usability of software products are hardly documented. The focus is more on collecting functional requirements [52]
Software qualification testing	Testing knowledge	This refers to the lack of proper testing knowledge [25]. This may be due to undefined process descriptions or quality manuals in small companies. This results in uncertainty regarding when and what to test
Software quality assurance	QA definition	The meaning of <i>QA</i> differs between companies [44]. The lack of proper process knowledge and a clear definition of <i>QA</i> may be the cause for this
	QA as overhead	Due to a smaller number of employees and resources, small companies consider <i>QA</i> activities as a burden [26]

(Continued)

Table 4. (Continued)

Process	Challenge	Description
Software documentation management	No documentation	The information is often transferred in discussions without documenting it. Due to this, scheduling a meeting at a specified time is a challenge in SSCs [25]
Software configuration management	CM guidelines.	Due to a lack of proper CM guidelines, there is uncertainty about which work products should be with version control [52]. This is due to an undefined process description or the lack of quality manuals in SSC
Life cycle model management	Avoid process adoption	SSCs usually avoid the adoption of any process standards in their software development process [14]. The reason could be due to that SSCs' consider process assessments and improvement as overly complex or think that they could raise project costs and delay project delivery

5 Discussion and Conclusion

In this paper, we have applied a systematic review method to analyze the literature related to processes within the SSC context, with the goal of exploring the state of the practice of the software development processes and the challenges associated with them.

In regard to RQ1: What is the state-of-practice in terms of the use of processes in small software companies? The current state of the practice related to the software development processes in SSCs can be seen in Table 3. It shows that software requirements analysis (49 %) within software implementation processes, project planning (39 %) within project processes, configuration management (22 %) in software support processes, and life cycle model management (56 %) in organizational project-enabling processes are the frequently considered processes for improvement when software process improvement (SPI) programs are conducted in SSCs. This suggests that SSC are mostly attracted to the above processes. The reason for the significant frequency of software requirement may be that requirements are considered an obligatory process and are often fundamental to any software company's success [51]. Therefore, it is considered to be a critical process for further improvements since SSCs wish to quickly deliver what their customers want to remain competitive in a dynamic market and to maintain a healthy relationship with their customers [15]. Thus, focusing more on requirement engineering seems to be legitimate, based on the nature of SSCs.

In regard to RQ2: What are the challenges that small software companies face in the processes? Several challenges were also found in the context of processes, such as a lack of understanding in terms of what collected requirements really mean, a lack of

testing knowledge, and a lack of CM guidelines (see Table 4). The reported challenges mostly fall under the frequently reported processes in Table 4. Typically, SSCs do not implement the necessary SPI programs for improving the process [15, 7]. In fact, most SSCs lack the required knowledge [14]. This could be a possible root cause for the many challenges revealed in the primary study papers.

Our observation from this review is that most of the studies we analyzed were conducted in collaboration with SSCs and that some also included medium-sized companies. Some papers referred to the SPI program based on models such as CMMI and SPICE within SSCs. We also observed that the SPI efforts carried out in SSCs are significantly reported in the literature, while descriptions of software processes in SSCs have received surprisingly little attention. An exploration of the empirical methods used in the primary studies showed that the majority of the studies were case studies. However, based on a systematic and validated model [19], our analysis shows that the majority of these papers fall under the low-rigor category (i.e., most of the studies lacked an adequate description of the study context, study design, and/or study validity). Therefore, further rigorous empirical studies within the context of software processes and SSCs are required. Various challenges were also found during the review that should be validated empirically in future research.

Acknowledgments. This research is supported by HILLA and ICT SHOK N4S (Need for Speed) programs funded by TEKES (Finnish Funding Agency for Technology and Innovation).

References

1. Arora, A., Gambardella, A.: The globalization of the software industry: perspectives and opportunities for developed and developing countries (2004)
2. Richardson, I., von Wangenheim, C.G.: Why are small software organizations different? *IEEE Softw.* **24**, 18–22 (2007)
3. Von Wangenheim, C.G., Weber, S., Hauck, J.C.R., Trentin, G.: Experiences on establishing software processes in small companies. *Inf. Softw. Technol.* **48**, 890–900 (2006)
4. Pettersson, F., Ivarsson, M., Gorschek, T., Öhman, P.: A practitioner's guide to light weight software process assessment and improvement planning. *J. Syst. Softw.* **81**, 972–995 (2008)
5. Fuggetta, A., Di Nitto, E.: Software process. In: *Proceedings of the on Future of Software Engineering - FOSE 2014*, pp. 1–12. ACM Press, New York, USA (2014)
6. De Castro, R.M., Braga, J.L., Soares, L.S., Oliveira, A.D.P.: Selection of software development good practices in micro and small enterprises: an approach using knowledge-based systems. In: *2012 31st International Conference on Chilean Computer Science Society*, pp. 12–20 (2012)
7. Pino, F.J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. *Softw. Qual. J.* **16**, 237–261 (2008)
8. Sulayman, M., Mendes, E.: A systematic literature review of software process improvement in small and medium web companies. In: Ślęzak, D., Kim, T.-H., Kiumi, A., Jiang, T., Verner, J., Abrahão, S. (eds.) *ASEA 2009. CCIS*, vol. 59, pp. 1–8. Springer, Heidelberg (2009)

9. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: a systematic mapping study. *Inf. Softw. Technol.* **56**, 1200–1218 (2014)
10. Klotins, E., Unterkalmsteiner, M., Gorschek, T.: Software engineering knowledge areas in startup companies : a mapping study. In: Fernandes, J.M., Machado, R.J., Wnuk, K. (eds.) *ICSOB 2015. LNBP*, vol. 210, pp. 245–247. Springer, Heidelberg (2015)
11. Zhang, H., Ali Babar, M.: Systematic reviews in software engineering: an empirical investigation. *Inf. Softw. Technol.* **55**, 1341–1354 (2013)
12. ISO/IEC/IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes (2008)
13. Espinosa-Curiel, I.E., Rodríguez-Jacobo, J., Fernández-Zepeda, J.A.: A framework for evaluation and control of the factors that influence the software process improvement in small organizations. *J. Softw. Evol. Process.* **25**, 393–406 (2013)
14. Basri, S., O'Connor, R.V.: Understanding the perception of very small software companies towards the adoption of process standards. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) *EuroSPI 2010. CCIS*, vol. 99, pp. 153–164. Springer, Heidelberg (2010)
15. Tosun, A., Bener, A., Turhan, B.: Implementation of a software quality improvement project in an SME: a before and after comparison. In: 2009 35th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2009, pp. 203–209 (2009)
16. O'Connor, R.V., Laporte, C.Y.: Using ISO/IEC 29110 to harness process improvement in very small entities. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) *EuroSPI 2011. CCIS*, vol. 172, pp. 225–235. Springer, Heidelberg (2011)
17. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. *Engineering* **2**, 1051 (2007)
18. StArt — LaPES - Laboratório de Pesquisa em Engenharia de Software. http://lapes.dc.ufscar.br/tools/start_tool
19. Ivarsson, M., Gorschek, T.: A method for evaluating rigor and industrial relevance of technology evaluations. *Empir. Softw. Eng.* **16**, 365–395 (2011)
20. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011)
21. Yin, R.K.: *Case Study Research: Design and Methods*. Sage Publications, Thousand Oaks (2009)
22. Hidayah, I., Wahyuni, W., Nugroho, L.E.: Process model and software process improvement for small software organization: an ethnographic study in Indonesia. **2**, 852–856 (2012)
23. Savolainen, P., Sihvonen, H.-M., Ahonen, J.: SPI with lightweight software process modeling in a small software company. *Softw. Process Improv.* **4764**, 71–81 (2007)
24. Ayalew, Y., Motlhala, K.: Software process practices in small software companies in Botswana. In: 2014 14th International Conference on Computational Science Its Applications, pp. 49–57 (2014)
25. Valtanen, A., Ahonen, J.J.: Big improvements with small changes: improving the processes of a small software company. In: Jedlitschka, A., Salo, O. (eds.) *PROFES 2008. LNCS*, vol. 5089, pp. 258–272. Springer, Heidelberg (2008)
26. Lester, N.G., Wilkie, F.G., Mcfall, D., Ware, M.P.: Investigating the role of CMMI with expanding company size for small- to medium-sized enterprises. *J. Softw. Maint. Evol. Res. Pract.* **22**, 17–31 (2010)
27. Jantunen, S.: Exploring software engineering practices in small and medium-sized organizations. In: Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2010, pp. 96–101 (2010)

28. Yilmaz, L., Phillips, J.: The impact of turbulence on the effectiveness and efficiency of software development teams in small organizations. *Softw. Process Improv. Pract.* **12**, 247–265 (2007)
29. O'Connor, R.V., Basri, S., Coleman, G.: Exploring managerial commitment towards SPI in small and very small enterprises. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) *EuroSPI 2010. CCIS*, vol. 99, pp. 268–279. Springer, Heidelberg (2010)
30. Jezreel, M., Mirna, M., Pablo, N., Edgar, O., Alejandro, G., Sandra, M.: Identifying findings for software process improvement in SMEs: an experience. In: *Proceedings of the 2012 9th Electronics, Robotics and Automotive Mechanics Conference CERMA 2012*, pp. 141–146 (2012)
31. von Wangeheim, C.G., Anacleto, A., Salviano, C.F.: Helping small companies assess software processes. *IEEE Softw.* **23**, 91–98 (2006)
32. von Wangenheim, C.G., Varkoi, T., Salviano, C.F.: Standard based software process assessments in small companies. *Softw. Process Improv. Pract.* **11**, 329–335 (2006)
33. Basri, S., O'Connor, R.: Organizational commitment towards software process improvement an Irish software VSEs case study. In: *Proceedings of 4th International Symposium on Information Technology 2010 (ITSim 2010)*, Malaysia, June 2010
34. Pino, F.J., Garcia, F., Piattini, M.: Key processes to start software process improvement in small companies. In: *Proceedings of the 2009 ACM Symposium on Applied Computing*, pp. 509–516 (2009)
35. Del Maschi, V.F., Spinola, M.M., Costa, I.A., Esteves, A.L., Souza, L.S., Vendramel, W., Pirola, J.: Practical experience in customization for a software development process for small companies based on RUP process and MSF. In: *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, pp. 71–93 (2008)
36. Laporte, C.Y., Alexandre, S., O'Connor, R.: A software engineering lifecycle standard for very small enterprises. In: O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.) *EuroSPI 2008. CCIS*, vol. 16, pp. 129–141. Springer, Heidelberg (2008)
37. Habra, N., Alexandre, S., Desharnais, J.M., Laporte, C.Y., Renault, A.: Initiating software process improvement in very small enterprises: experience with a light assessment tool. *Inf. Softw. Technol.* **50**, 763–771 (2008)
38. Nilsson, A., Castro, L.M., Rivas, S., Arts, T.: Assessing the effects of introducing a new software development process: a methodological description. *Int. J. Softw. Tools Technol. Transf.* **17**, 1–16 (2013)
39. McCaffery, F., McFall, D., Wilkie, F.G.: Improving the express process appraisal method. In: Bomarius, F., Komi-Sirviö, S. (eds.) *PROFES 2005. LNCS*, vol. 3547, pp. 286–298. Springer, Heidelberg (2005)
40. Hauck, J.C.R., von Wangenheim, C.G., de Souza, R.H., Thiry, M.: Process reference guides – support for improving software processes in alignment with reference models and standards. In: O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.) *EuroSPI 2008. CCIS*, vol. 16, pp. 70–81. Springer, Heidelberg (2005)
41. Cater-Steel, A.P.: Low-rigour, rapid software process assessments for small software development firms. In: *Proceedings of the Australian Software Engineering Conference, ASWEC 2004*, pp. 368–377 (2004)
42. Hurtado, J.A., Bastarrica, M.C., Ochoa, S.F., Simmonds, J.: MDE software process lines in small companies. *J. Syst. Softw.* **86**, 1153–1171 (2013)
43. Tuffley, A., Grove, B., McNair, G.: SPICE for small organisations. *Softw. Process Improv. Pract.* **9**, 23–31 (2004)
44. Wilkie, F.G., McFall, D., McCaffery, F.: An evaluation of CMMI process areas for small - to medium-sized software development organisations. *Softw. Process Improv. Pract.* **10**, 189–201 (2005)

45. Diaz, J., Garbajosa, J., Calvo-Manzano, J.A.: Mapping CMMI level 2 to scrum practices: an experience report. In: O'Connor, R.V., Baddoo, N., Cuadrado Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 93–104. Springer, Heidelberg (2009)
46. Clarke, P., O'Connor, R.V.: An empirical examination of the extent of software process improvement in software SMEs. *J. Softw. Evol. Process* **25**(9), 981–998 (2013)
47. Caballero, E., Calvo-Manzano, J.A., San Feliu, T.: Introducing scrum in a very small enterprise: a productivity and quality analysis. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 215–224. Springer, Heidelberg (2011)
48. O'Connor, R.V.: Evaluating management sentiment towards ISO/IEC 29110 in very small software development companies. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 277–281. Springer, Heidelberg (2012)
49. Ribaud, V., Saliou, P., O'Connor, R.V., Laporte, C.Y.: Software engineering support activities for very small entities. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 165–176. Springer, Heidelberg (2010)
50. Allison, I.: Organizational factors shaping software process improvement in small-medium sized software teams: a multi-case analysis. In: Proceedings of the 7th International Conference on Quality of Information and Communications Technology, QUATIC 2010, pp. 418–423 (2010)
51. McCaffery, F., Taylor, P.S., Coleman, G.: Adept: a unified assessment method for small software companies. *IEEE Softw.* **24**, 24–31 (2007)
52. Saastamoinen, I., Tukiainen, M.: Software process improvement in small and medium sized software enterprises in eastern finland: a state-of-the-practice study. In: Dingsøyr, T. (ed.) EuroSPI 2004. LNCS, vol. 3281, pp. 69–78. Springer, Heidelberg (2004)
53. Zhang, Y., Zhao, X., Zhang, X., Zhang, T.: Test effectiveness index: integrating product metrics with process metrics. In: 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER 2012, pp. 54–57 (2012)
54. Alexandre, S., Renault, A., Habra, N.: OWPL: a gradual approach for software process improvement in SMEs. In: Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2006, pp. 328–335 (2006)
55. Basri, S., O'Connor, R.V.: Towards an understanding of software development process knowledge in very small companies. In: Abd Manaf, A., Sahibuddin, S., Ahmad, R., Mohd Daud, S., El-Qawasmeh, E. (eds.) ICIEIS 2011, Part III. CCIS, vol. 253, pp. 62–71. Springer, Heidelberg (2011)
56. Suula, M., Makinen, T., Varkoi, T.: An approach to characterize a software process. In: 2009 Portland International Conference on Management of Engineering and Technology, PICMET 2009, pp. 1103–1109 (2009)