The state of research on software engineering competencies: A systematic mapping study☆

Nana Assyne [a],[*], Hadi Ghanbari [b],[c], Mirja Pulkkinen [a]

[a] *University of Jyväskylä, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 Jyväskylä, Finland*
[b] *Aalto University, School of Business, Ekonominaukio 1, 02510, Espoo, Finland*
[c] *FinEst Centre for Smart Cities, TalTech, Estonia*

ARTICLE INFO

ABSTRACT

Considering the critical role of software in modern societies, we face an urgent need to educate more competent software professionals. Software engineering competencies (SEC) are considered the backbone of successfully developing software products. Consequently, SEC has become a hotspot for software engineering research and practice. Although scientific literature on SEC is not lacking, to our knowledge, a comprehensive overview of the current state of SEC research is missing. To that end, we conducted an extensive and systematic review of the SEC literature. We provide an overview of the current state of research on SEC, with a particular focus on common SEC research areas. In addition to reporting the available SEC models and frameworks, we compile a list of 49 unique essential competencies of software professionals. Finally, we highlight several gaps in the literature that deserve further research. In particular, we call for a better understanding of how the essential competencies of software professionals change over time, as well as fresh accounts of the essential competencies of software professionals. Additionally, considering recent shifts toward Agile and DevOps methods, future research must explore the competencies required for developing software products in modern development environments.

1. Introduction

Software drives or affects every device we use for any area of our lives. Developing software products is a complex and resource-intensive process that requires competent software professionals (Casale et al., 2016). However, the software industry faces an increasing demand for, and a significant shortage of skilled software professionals (Calazans et al., 2017; Gimenes et al., 2012). Considering this shortage and the critical role of software in modern life, there is an urgent need to educate and train more competent software professionals. Generally speaking, competence is defined as "a collection of skills, abilities and attitudes to solve a problem in a given context" (Holtkamp et al., 2015, p. 137). In software engineering, competence covers the entire development process and entails software professionals' knowledge, skills, and attitudes to fulfill a task in a software development project (IEEE, 2014).

Software engineering competencies (SEC) have become a strategic research area for the software engineering discipline (Colomo-Palacios et al., 2013), and much literature has been published in this area (Lenberg et al., 2015). However, to our knowledge, the current literature lacks a comprehensive overview of SEC research. Instead, previous studies only explored specific areas in SEC, such as the role of personality in software engineering (Cruz et al., 2015; Moustroufas et al., 2015; Vishnubhotla et al., 2018), measurement of software engineers' capability and competence (Vishnubhotla et al., 2018), and models for recruiting software professionals (Moustroufas et al., 2015). Hence, a study that comprehensively overviews the current state of the research on SEC and identifies the gaps requiring future studies is needed. Additionally, considering the emergence of various software development approaches during the last three decades, it would be beneficial to understand how essential SEC have changed and evolved over the years. To that end, in this paper, we aim to answer the following research question: "*What can the current literature inform us about the evolution of essential SEC?*"

To answer this research question, we conducted a systematic mapping study. A mapping study is suitable for comprehensively overviewing the literature in a topic area where numerous papers exist (Kitchenham & Charters, 2007; Li et al., 2015; Petersen et al., 2008). Since we aimed at comprehensively overviewing SEC topic areas and considering the richness of the literature on this topic, a mapping study was conducted. Through an extensive

search and rigorous evaluation process, 60 primary studies were identified for this review. By analyzing these primary studies, we overviewed the current state of research on SEC, focusing on common SEC research areas, available SEC models or frameworks, and the essential competencies of software professionals. In particular, we show that previous literature on SEC focuses on the two main areas of personnel and organizational SEC. Additionally, we identified 14 models or frameworks that can be used for identifying, assessing, and developing SEC. Furthermore, we identified 49 essential competencies and classified them into 11 themes.

Our results indicate that despite a rich body of literature, several areas of SEC need further scientific investigation. First, future studies are needed to propose better models or frameworks for providing theoretical accounts and practical implications on different aspects of SEC, especially competence assessment in various software engineering tasks and discerning different levels of satisfaction. Additionally, future empirical research is needed to better understand how software professionals' competencies change over time or as they move from one role to another. Finally, further research is needed to assess and provide a fresh account of the essential competencies of future software professionals, especially regarding modern development methods and techniques, such as the agile approach (Abrahamsson et al., 2002) and DevOps (Debois, 2011).

The rest of this paper is structured as follows: Section 2 describes the research background, Section 3 outlines the research method used, Section 4 looks at the results, Section 5 discusses the results and research gaps, and Section 6 presents the conclusions.

2. Research background

The words "competence" or "competency" are often used interchangeably and sometimes have varied and ambiguous meanings, depending mainly on the content or cultural contexts (Le Deist and Winterton, 2005). To avoid such conceptual ambiguities, we first need to define our key concepts and overview the context in which this study can be viewed.

2.1. Concepts and definitions

In their study, Sedelmaier and Landes (2014b) state that *"Competence denotes a comprehensive capability to act appropriately in complex situations. The capability to act includes technical knowledge, also called factual knowledge. The capability to cope with complex and new situations also presupposes additional skills, which are often subdivided into social, personal and methodological competence"*. Adopting this definition, we define competence in software engineering as *"a complete set of abilities, skills, knowledge, and capabilities needed to activity engage in a software development effectively"*.

As such, competencies are associated with both individuals and enterprises. From the individual perspective, they are related to education and human resource management (Le Deist and Winterton, 2005). However, from the enterprise perspective, they are concerned with the resource-based view of the firm (Wernerfelt, 1984). Additionally, competencies can be divided into two main types: hard and soft competencies (Harzallah and Vernada, 2002; Havelka and Mermout, 2009; Rainsbury et al., 2002; Tahvanainen and Luoma, 2018). "Soft skills are personal individuality that primarily impact the behavior of a person while interacting with others in a working setup" (Ahmed et al., 2013, p. 172). In contrast, hard competencies are teachable skills required for job performance or assignment and are acquired mostly through formal training and studies (Urs, 2013). Often, the trainee is

required to be smart or possess a good IQ to acquire the required skill (Andrews and Higson, 2008; Harzallah et al., 2002; Trivellas and Reklitis, 2014).

Sedelmaier and Landes (2013) argued that teaching software engineering is difficult and cookbooks to ease this difficulty are lacking. Thus, the competencies needed in software engineering require identification. Given this, the authors set up an SEC research agenda to identify the competencies needed by software professionals using an experiential learning process by observing some students. Using the Software Engineering Body of Knowledge (SWEBOK), Sedelmaier and Landes (2014b) identified competencies needed by developers and proposed the Software Engineering Body of Skills (SWEBOS). The authors concluded that technical skills (i.e., hard competencies) are well noted in the literature, but the same is not true for non-technical skills (i.e., soft competencies). Subsequently, they created a framework for identifying the non-technical skills required from software professionals (Sedelmaier and Landes, 2014a).

Using interviews with 23 managers and hiring personnel from the USA and Europe, Studt et al. (2015) investigated what recent graduates struggle with when they start to work and what deters them from gaining employment. They reported that recent graduates struggle with several soft and hard skills (e.g., effective communication, using configuration management systems and other tools, product unit testing). However, a lack of project experience and problem-solving abilities mostly prevents them from getting employed (Radermacher et al., 2014). Thurner et al. (2016) used their experiences in the SEC research area and identified some reasons why students are struggling with their soft skills and competencies during the initial stages of their studies: (i) heterogeneity of the initial skill to study, (ii) lack of clues on how to assess their own competencies, (iii) lack of clues on how to expand their competencies, and (iv) lecturers mostly focus on teaching technical skills and neglect soft skills. Therefore, they designed a program to improve the situation of freshmen in an integrated form (Thurner et al., 2016).

In organizational settings, competencies are linked to roles. Hence, we consider the roles and tasks or responsibilities associated with software engineering. Depending on the organization, different terms may be associated with certain roles (e.g., software developer, software designer, software engineer). This makes the study of the competencies of these roles difficult. To avoid such confusion, in this study, we use the term "software professional" to discuss individuals who employ the necessary skills to design, construct, test, and maintain computer software (Kalliamvakou et al., 2019; Kobata et al., 2013; León-sigg et al., 2018). Therefore, we used the roles defined in SWEBOK (Bourque and Fairley, 2014) and the software engineering competency model SWECOM (IEEE, 2014). Thus, we listed some of the different roles and their associated positions and tasks or responsibilities for a typical software development project (Table 1).

SWECOM, inspired by SWEBOK, provides various knowledge and skill sets for software engineering. These skill sets are grouped into five areas: cognitive skills, behavioral attributes and skills, requisite knowledge, related disciplines, and technical skills. SWECOM further divides technical skills into life cycle skill areas and cross-cutting skill areas. While cross-cutting skill areas are skill sets needed throughout the life cycle of developing software, life cycle skill areas are those needed to perform a task in different phases of software development. Thus, since this study aims to identify competencies and tasks associated with them and relate them to software development roles, we used the life cycle skill (comprising skills and activities) and mapped it to the software engineering roles in SWEBOK.

Table 1 shows that a role can be associated with different positions, and it may have different tasks or responsibilities. Additionally, different positions are given to software professionals

Table 1
Roles, associated positions, and tasks of software professionals.

Roles	Description	Associated positions	Tasks
Requirement analyst	The role represents software professional(s) responsible for eliciting the functional and non-functional attributes of a software system to meet the goals of a customer.	System engineer, requirement technical leader, requirement engineer, requirement analyst, lead requirement manager, requirement technician, etc.	Identify stakeholders, perform analysis on the requirement, use appropriate means to describe the requirements, construct, and analyze prototypes, etc.
Designer	The role represents software professional(s) responsible for designing software architecture based on the requirement.	Software designer, lead designer, technical designer, design technician, etc.	Design technics for software design, manage software design activities, specify a common interface, and use software design review, etc.
Programmer	The role represents software professional(s) responsible for constructing the software and converting the design specifications into functional software.	Senior software engineer, lead developer, technical software leader, software technician, etc.	Select the environment for developing the software, monitor the software development process, create code, and implement the design, and document and comment on the code, etc.
Software test & quality engineer	The role represents software professional(s) responsible for overseeing the testing and quality issues of software.	Senior test/quality engineer, lead software test/quality engineer, software test/quality engineer, test/quality engineer, test technician, etc.	Identify stakeholders and tools for testing the software, develop a test plan for testing the software, and collect and report data resulting from testing/demonstration, etc.
Configuration & maintenance engineer	The role represents software professional(s) responsible for maintaining and sustaining the software during its life cycle.	Senior maintenance/ configuration engineer, lead maintenance/ configuration engineer, maintenance/ configuration engineer, maintenance technician, etc.	Develop transition and identify stakeholders for transition, maintain software configuration, perform problem identification and correction, and monitor and analyze software maintenance activities, etc.

depending on the organization's size or the type of software project to be undertaken (Saiedian and Dale, 2000). Similarly, the tasks associated with these roles can be performed by a dedicated person or shared and performed by all team members. The latter especially applies to agile software development methodologies, which rely on small and self-organized teams. In the past few decades, software engineering has been undergoing a paradigm shift with the agile approach (Abrahamsson et al., 2002); (Dyba and Dingsoyr, 2008). Regarding the roles and responsibility areas, we are aware of this change and new role definitions emerging with it, such as "product owner" and "scrum master". However, as our study shows, even if agile methods are broadly adopted in practice and are also present in software engineering research, their specificity has not yet received much attention in SEC research. Therefore, we rely on the traditional role definitions as the basis for our competence study.

2.2. Related literature review studies

Three literature review studies on SEC were found from the literature search. Cruz et al. (2015) used a systematic mapping study to plot the current landscape of published empirical and theoretical studies that explored the role of personality in software engineering. The authors reviewed 90 papers published from 1970 to 2010. The study identified the Myers–Briggs Type Indicator as the regularly used test for accessing personality profiles in software engineering, and pair programming and education were identified as the two popular topics being researched. The breakdown of the topics is as follows: pair programming, education, team effectiveness, software process allocation, software engineer personality characteristics, and individual performance, comprising over 88% of the studies. Less frequently researched topics include team process, behavior and preferences, and leadership performance, and they comprise the rest of the studies. The authors argued that although the number of published articles in the personality research area in software engineering has grown in recent years, the evidence found to support this growth was inconclusive in their study. However, they suggested that the

research community could improve and extend the findings in the field to include the practitioners' characteristics (Cruz et al., 2015).

Another study by Vishnubhotla et al. (2018) utilized a systematic literature review to present the state of the art relating to the capability and competence measurement of software professionals, including teamwork in agile software development. For this study, 16 related articles were selected. Evidence of attributes relevant and appropriate to measuring the capability of software engineers and teams was reported. Furthermore, their findings identified different instruments for measuring these attributes. The findings further identified the effective categorization of these attributes into groups for project managers to use as a checklist when allocating software engineers into teams (Vishnubhotla et al., 2018).

Finally, Moustroufas et al. (2015) employed a traditional literature review to evaluate the adequacy of software professional competencies concerning a software competence profiling model for recruiting software professionals. According to the authors, the model review helps in the recruitment process of software professionals for software development. Regarding the review, they suggested a new customizable, three-level model designed based on the standards of a generic framework developed by the European project ComProFITS for recruiting software professionals (Moustroufas et al., 2015).

The articles by Cruz et al. (2015) and Vishnubhotla et al. (2018) utilized a rigorous research methodology but focused on some aspects of SEC research areas, such as personality in software engineering and capability measurement of software professionals in agile software development projects; therefore, they did not comprehensively overview the SEC research area. Moustroufas et al. (2015) created a profiling model to assist in recruiting software professionals. However, they did not employ a systematic approach to extricate their model. Thus, their model lacks scientific rigidity in its development. Despite these limitations, the results present a productive theme for the SEC research area. Given the aforementioned limitations in the three review studies on SEC, this paper aims to propose a comprehensive state-of-the-art overview of SEC research.
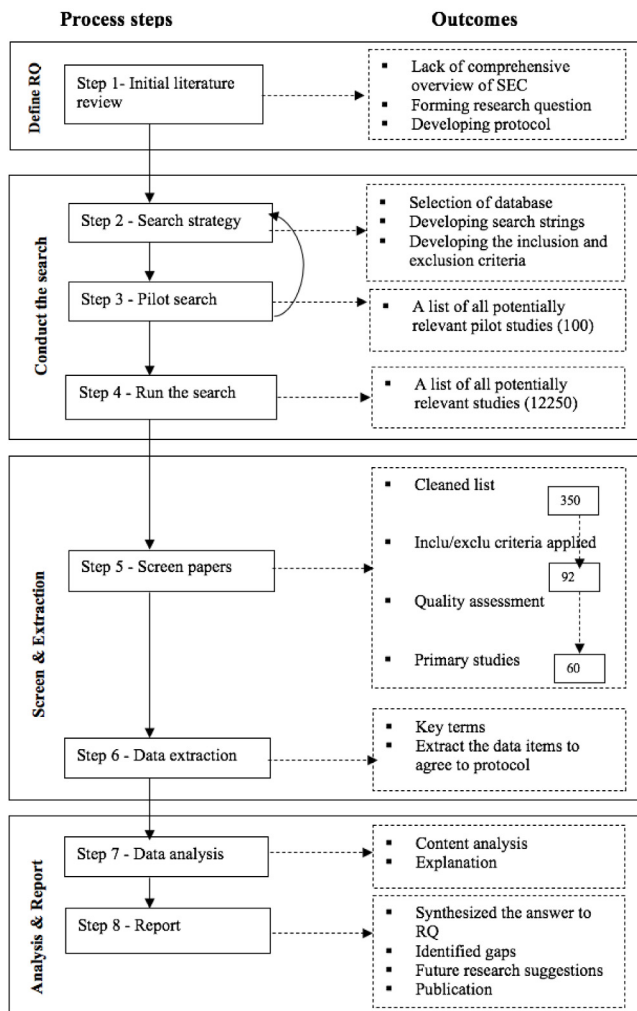
Fig. 1. The research process followed in this study.

3. Research methodology

An extensive literature review was conducted following the study objective. Fink (2010, p. 3) defined a literature review as "a systematic, explicit and reproducible method for identifying, evaluating and synthesizing the existing body of completed and recorded work produced by researchers, scholars, and practitioners". Two types of literature reviews were recognized, and one was selected for the purpose of this study. A systematic literature review is defined as "a form of secondary study that utilises a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable" (Kitchenham and Charters, 2007, p. vi). It is utilized to summarize an existing concern and provide a background for positioning new research. A systematic mapping study has the goal "to overview a research area and identify the quantity and type of research and results available within it" (Petersen et al., 2008, p. 2). Thus, since this study primarily overviews the SEC research area, the authors employed a systematic mapping study. Accordingly, the authors are guided by the guidelines and principles of Petersen et al. (2008, 2015). Fig. 1 depicts the four stages of the mapping study adopted from Petersen et al. (2008, 2015).

Fig. 1 shows the four stages of the research process, which include defining the research question, conducting the search, screening papers and data extraction, and analyzing and reporting

the results of the review. Each stage and its underlying steps are explained in the following subsections.

3.1. Stage one—defining the research questions

This section presents Step 1, referred to in Fig. 1 as an initial literature review. Although the initial review shows abundant literature on SEC, it lacks a comprehensive overview of the research area. Therefore, the authors set the objective of providing a comprehensive overview of the current state of research on SEC and identifying potential gaps for future studies. Thus, the authors set the main research question as "*What can the current literature inform us about the evolution of essential SEC?*"

To answer this research question, we tried to approach the body of literature on SEC with a particular focus on different research areas on SEC, the frameworks and models for studying SEC, the essential competencies for software professionals, and SEC evolution over time. Therefore, we formulated the following sub-questions:

RQ1—What research areas are emerging in the body of literature about SEC?

RQ2—What SEC models or frameworks have been developed?

RQ3—What are the essential competencies of software professionals?

RQ4—How has SEC research evolved over the last three decades?

3.2. Stage two—Conducting the search

This stage involves Steps 2 to 4 of Fig. 1. Step 2 is the search strategy. In this step, the electronic databases for this study were selected by all the authors. To ensure wider coverage of the extant literature from software engineering and its neighboring disciplines, including information systems and computer science, we decided to search IEEE Xplore, ACM Digital Library, Scopus, AIS eLibrary, and ScienceDirect databases.

After selecting these databases, search strings were developed based on the population and intervention strategies suggested by previous studies (Petersen et al., 2008, 2015; Kitchenham and Charters, 2007). Considering the subject area—software engineering competencies—the population of our study is software engineering, and the intervention is competence, competency, or competencies (i.e., competenc*). Regarding the population, while the software engineering literature was our main target, information systems (IS) and computer science (CS) were also considered relevant fields. Since IS and CS studies often use terms such as software development or systems development instead of software engineering, we decided to consider all these variants in our population. Regarding the intervention, the term competence or competency can be denoted with terms such as skills, attitude, abilities, knowledge, etcetera. Therefore, while developing and testing our search strings, we considered these terms. However, the search results showed that adding such general terms to the string induced too many irrelevant studies. As such, since we aimed to cohesively depict the literature on SEC, it is plausible to assume that any previous study on the subject area at least mentions competenc* somewhere in the body of the paper. Therefore, we made an informed decision to use the term competenc* to define the scope of our study. Consequently, we formed the following search string:

(("software develop* OR "system* develop*" OR "software engineer*") AND (competenc*)).

It is important to note that, considering the differences between electronic databases, we made adjustments to the above search string to fit each database. Before starting to search for

Table 2
Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
• Studies reported in the English language. • The study must be peer-reviewed (journal, conference, or workshop). • The study focuses on software development competencies. • The study answers one of the research questions.	• Studies that do not fulfill every inclusion criterion. • Studies in which access to the full text is not possible. • Studies in which access to the digital version is not possible.

relevant studies, the authors also developed a set of inclusion and exclusion criteria (Table 2).

In Step 3 of the process, a pilot search was conducted using the Scopus database to test the search strings and also for validation. The quality of the search string is essential for any systematic review; therefore, it is important to ensure that steps are taken to avoid retrieving too much information or too little information for the study (Cruz et al., 2015). Following the guidelines of Kitchenham and Charters (2007) and Petersen et al. (2008, 2015), steps were taken to improve the search string to ensure that it retrieved regulated information. Thus, different search strings were tested using the Scopus database (see Appendix A). As argued by Cruz et al. (2015), the search string quality can be viewed from two indexes: the string sensitivity and precision. Sensitivity refers to the ratio between the number of relevant studies retrieved and the total number of existing relevant studies in the literature. Precision is the ratio between the number of relevant studies retrieved and the total number of studies retrieved. Following Cruz et al. (2015), it would be ideal if both indexes could be achieved. However, in practice, that may not be the case; therefore, in selecting a string for a review, there should be a trade-off. Thus, the authors tested different search strings by considering the sensitivity index, and the first 100 items were selected to check with known literature in the area (e.g., Stepanenko and Kashevnik, 2017; Vishnubhotla et al., 2018) to validate the search term or terms to be used. After testing various search strings shown in Appendix A, the authors agreed to use the string stated above as appropriate for this study.

In Step 4, the authors ran the full search. In performing the search, the authors did not set a lower bound date for the search because they wanted to have wider coverage. However, the upper bound date was on the day of the search. Here, the retrieval of the meta-data for the study was February 15, 2019 for all the databases. Table 3 shows the number of items or papers retrieved using the automatic search done by adapting the search strings to the search engine of each database. A total of 12,250 potentially relevant papers were retrieved from all five databases. The authors stored the meta-data of the papers in a spreadsheet.

3.3. Stage three—Screening papers and data extraction

The third stage of the research process comprises screening (i.e., Step 5) and data extraction (i.e., Step 6) (Fig. 1). In Step 5, the authors combined all the potentially relevant papers from the different databases into one spreadsheet. Duplicates, studies without titles, studies without abstracts, research in progress, and posters were eliminated to generate a clean list. Next, the first author evaluated all the potentially relevant studies (n = 12,250) and applied the inclusion/exclusion criteria stated in Section 3.2. This was done by first reading the titles and abstracts and categorizing the studies as "relevant",[1] "not relevant",[2] "maybe relevant",[3] or "cannot decide".[4] However, in cases where the

relevance of a paper could not be evaluated based on its title and abstract, the author read the introductions, contributions, and conclusions of the paper. To avoid individual mistakes and bias, after this round was completed, the second author randomly picked 100 papers from the list of 346 papers that were in the category of "relevant" to check if the inclusion/exclusion criteria applied as stated. The second author further read through all 51 papers that were in the categories of "maybe relevant" and "cannot decide" and evaluated each paper following the inclusion/exclusion criteria. Next, the authors discussed their individual evaluations of these 51 papers and agreed to categorize four papers as relevant and exclude 47 papers. Next, the first author read each of these 350 papers fully and applied the inclusion/exclusion criteria.

As presented in Table 3, after cleaning the list and applying the inclusion and exclusion criteria, 92 papers were left for the next step in which the authors applied the quality assessment criteria. Kitchenham and Charters (2007) specified that in addition to the inclusion and exclusion criteria, quality assessment is needed to assess the quality of the original papers selected for the study. Authors such as Berg et al. (2018) and Molléri et al. (2018) suggested that quality assessment is not necessarily required for a systematic mapping study. However, it adds value to the outcome of mapping studies. Thus, in this mapping study, the authors adopted the 11-point quality criteria of Dyba and Dingsoyr (2008), which relate to the rigor, credibility, and relevance of research for the study (see Table 4). The authors used "yes = 1" or "no = 0" in response to each question regarding the quality criteria. After applying the quality assessment criteria, 60 papers qualified for the review, see Appendix B.[5] In Step 6 (i.e., data extraction), the authors read the 60 primary studies in full detail and coded the contents of the selected papers that are related to our research questions into a spreadsheet.

It must be noted that the criteria proposed by Dyba and Dingsoyr (2008) were originally designed to assess the quality of empirical studies. However, Table 4 shows that eight criteria apply to both empirical and conceptual papers, and only three of them concentrate on empirical studies (i.e., Criteria 3, 5, and 6). Additionally, Criterion 5 is concerned with the rigor of the sampling process. Therefore, in conceptual papers where the unit of analysis can be a study (i.e., a case), sampling refers to the process of selecting previously published studies. Therefore, nine out of the eleven criteria suggested by Dyba and Dingsoyr (2008) could be used to assess the quality of both empirical and conceptual studies. Therefore, to ensure the reliability of the primary studies, we decided to exclude those studies that failed to meet at least four of the eleven criteria (i.e., a 33% threshold when only considering the nine criteria that apply to both empirical and conceptual studies).

3.4. Stage four—Analyzing and reporting

In Step 7, data analysis was performed on the extracted primary studies to answer the research questions. To analyze the

---

[1] Relevant—papers that the first author is 100% sure about their relevance

[2] Not relevant—papers that the first author is 100% sure that they are not relevant

[3] Maybe relevant—papers that the first author considers relevant, but he is not 100% sure about their relevance

[4] Cannot decide—papers that the first author cannot decide if they are relevant or not

[5] The full list of the 92 papers and the result of the assessment used in the quality assessment can be found online on the link https://docs.google.com/spreadsheets/d/1afF5sIMogZdlJEzq3ap3RPgpNR2eJ5yjnaK_YV79l10/edit?usp=sharing.

Table 3

Results of screening papers.

Database	Automatic search result (Step 1)	Apply inclusion/ exclusion criteria	Apply quality assessment	Final primary papers
IEEE	2,596			
ACM	1,226			
ScienceDirect	3,986	350	92	60
AIS	2,725			
Scopus	1,717			
Total	**12,250**	**350**	**92**	**60**

Table 4

Quality assessment questions adapted from Dyba and Dingsoyr (2008).

No.	Quality question	Value in the criteria
1	Is this a research paper, or is it merely an experience report or expert opinion?	Research
2	Is there or are there a clear statement(s) of the aims of the research?	Aim
3	Is there an adequate description of the context in which the research was carried out?	Context
4	Was the research design appropriate to address the aims of the research?	Design
5	Was the sampling strategy appropriate to address the aims of the research?	Sampling
6	Was there a control group with which to compare treatments?	Control
7	Was the data collected in a way that addressed the research issue?	Data collection
8	Was the data analysis sufficiently rigorous?	Rigor
9	Has the relationship between the researcher and participants been considered to an adequate degree?	Reflexivity
10	Is there a clear statement of the findings?	Finding
11	Is the study of value for research or practice?	Value

Table 5

Contribution types.
Source: Adapted from Shaw (2003) and Theisen et al. (2017).

Contribution type	Description
Frameworks, methods, techniques	A particular framework, method, or technique that can be used to identify, assess, and develop SEC
Guidelines	List of advice or recommendations based on the synthesis of the obtained research results
Lessons learned	The set of outcomes directly based on the research results obtained from the data analysis
Model	The representation of an observed reality in concepts or related concepts after a conceptualization process
Tools	A technology, program, or application that is developed to support different aspects of software engineering
Advice/implications	A discursive and generic recommendation based on personal opinion

data extracted, we employed content analysis (Vaismoradi et al., 2013). According to Vaismoradi et al. (2013), content analysis is well suited for analyzing multifaceted data by labeling relevant items (coding) in the text and interpreting the content. The coding procedure also assists in quantifying the qualitative analysis results (e.g., counting the frequency of occurrence), which can indicate the significance of an issue. Content analysis is defined as a "systematic coding and categorising approach used for exploring large amounts of textural information unobtrusively" (Vaismoradi et al., 2013, p. 400). Hsieh and Shannon (2005) further discussed different approaches for conducting content analysis—conventional or open content analysis, directed content analysis, and summative content analysis. To analyze the extracted data, we combined these approaches. Conventional content analysis aims at an open and data-driven approach to describe a phenomenon, with no prior theory or framework guiding the analysis. For RQ1, aimed at identifying research areas in SEC studies for future directions, this content analysis approach was followed. The emerging topical areas were labeled and grouped following the research foci and outcomes (i.e., research contribution types). After the procedure, to classify the contributions of the primary studies, the extracted data were coded using the contribution types suggested by Shaw (2003) (Table 5).

RQ2 aimed to identify the available models and frameworks for studying SEC. Hence, the directed content analysis was applied to identify such items from the primary studies. The directed content analysis was led by the concept of a 'model' or 'framework' for SEC or some aspect of them to identify from prior research. For RQ3, to help in structuring the essential SEC for software development and training, the essential competencies of software professionals were extracted from the primary studies and tabulated to form categories. It was done without regarding

existing concepts, thereby following the conventional or open content analysis approach, so that the categories emerged from the data. The identified categories were then mapped to Rivera-Ibarra et al.'s (2010) framework. This framework provides a simplistic, granular overview of identifying different SEC as having technical or non-technical competence.

Literature on software engineering competence frameworks or models is not necessarily lacking. Specifically, frameworks or models for classifying and observing software competence have been studied and created (Acuña and Juristo, 2004; Acuña et al., 2006; Bröker, 2014; Rivera-Ibarra et al., 2010; Thurner et al., 2016; IEEE, 2014). In all these frameworks, observation and classification can be done from the traditional perspective (i.e., from a behavioral and technical perspective). However, Rivera-Ibarra et al. (2010) refined this granularity. That is, from a technical perspective, they further used the roles in software engineering. By far, those who try to extend it seem only to focus on the granularity of the traditional perspective. Therefore, the Rivera-Ibarra et al. (2010) framework is an obvious choice for our study since we aimed to identify competencies regarding the roles of software engineering.

Concerning RQ4, to identify trending changes in SEC research over the past three decades, we followed the directed and summative content analysis modes, resulting first in an overview (Section 4.1), presenting general facts about SEC studies published. Further, an analysis of these studies in Section 4.2 and the studies on essential competencies in Section 4.3 map the results to emerging research areas. As a final step (Step 8), we synthesized the answers to the research questions stated in Section 3.1. Based on the answers found, we further identified gaps in the existing knowledge and, accordingly, suggested future research. Each step of the research process in this study was iterated

Table 6
Publication channels of the primary studies.

Channels	Total publications	Percentage (%)
Conference	44	73
Journal	16	27
Total	**60**	**100**

several times. The first author handled the first round of each step in the research process, and the second author independently verified each step in the process. When there were disagreements on the result, all authors resolved it through discussion.

4. Results

This section presents the results of the analysis of the 60 primary studies in this research. The results serve as a state-of-the-art overview of the SEC research area to answer our research questions. Section 4.1 explores the overview of the primary studies, and Sections 4.2–4.5 involve the individual research questions.

4.1. Overview of primary studies

As shown in Table 3 of Section 3.3, the automatic search of the combined five databases resulted in 12,250 potentially relevant papers. After applying the inclusion and exclusion criteria and the quality assessment criteria, 60 primary studies were selected for the data analysis. Table 6 presents the publication channels of the primary SEC studies. Out of the 60 primary studies, 44 (73%) were conference publications, and 16 studies (27%) were journal publications.

Table 7 shows an overview of where the primary studies were published. Regarding scientific journals, the Journal of Systems and Software with three articles, followed by the Institution of Engineering and Technology Software, and Information and Software Technology, with two articles each, are at the top of the list. The remaining journals had one article each.

Regarding the conferences publishing SEC studies, the full details of all the channel outlets can be found in Appendix C. There were 30 conferences in total. The IEEE Global Engineering Education Conference published nine studies (P05, P13, P16, P23, P24, P26, P39, P56, and P57), followed by Frontiers in Education with three papers (P05, P25, and P35). Two papers each were published by the ACM Conference on Innovation and Technology in Computer Science Education (P14 and P54), the IEEE International Conference on Industrial Engineering and Engineering Management (P06 and P60), the International Conference on Product-Focused Software Process Improvement (P04 and P51), and the International Conference in Software Engineering Research and Innovation (P27 and P30). The remaining outlets, as presented in Appendix C, published one paper each.

Table 8 presents the types of research methods utilized by each of the primary studies. Out of the 60 studies, 46.7% utilized mixed methods, 35% employed a qualitative method, and 18.3% utilized a quantitative method.

Fig. 2 presents the distribution of the research methodologies used in the primary studies between 1992 and 2018. As can be seen, before 2010, there were very few studies. However, in recent years, there has been increasing interest in SEC research.

Table 9 presents an analysis of the contributions of the primary studies. In 42 out of the 60 primary studies (i.e., 70%), the main contribution was a set of lessons learned. The second highest contribution type was a model (13.3%), followed by frameworks (10%) and guidelines (5%), whereas only one primary study (1.7%) proposed a tool.

4.2. RQ1—What research areas are emerging in the body of literature about SEC?

RQ1 aimed at categorizing the SEC research areas. As stated in Section 3.4, to extract the focus of each study, all the selected primary studies were read in detail. Two main research areas were identified: *personnel competence* and *organizational competence*. Table 10 below overviews the research areas, comprising the abovementioned main research areas, their subareas with descriptions, and the primary studies categorized under these.

The personnel competence area comprises research that focuses on software professional competencies, i.e., the skills, abilities, and attitudes required for developing software products or services. These are the catalyst for developing a software product or service, and they include the soft and hard competence areas (e.g., creative thinking and programming skills). The organizational competence area focuses on tools or instruments such as assessment and identification models and frameworks, which are used for organizing, assessing, measuring, and managing personnel competencies.

As shown in Table 10, half of the primary studies focused on the personnel competence research area, and the other half focused on the organizational competence research area. In the areas emerging as subcategories of the main areas, the soft/hard competence research area had the highest number of studies, with 18 primary studies, followed by assessment/measurement, with 15 studies. Competence management had the third-highest number, with eight studies. Seven studies focused on competence identification. Six studies each focused on hard or soft competence areas separately.

A major development observed in the recent past in the software engineering area is transitioning to agile development methods (Abrahamsson et al., 2002). Therefore, an essential aspect of investigating the competencies in software engineering is to understand what this shift in the development paradigms brings regarding the competencies of software organizations and individuals in this profession. It must be noted that in our review, only 12 primary studies (P10, P11, P13, P20, P32, P41, P43, P48, P49, P52, P55, and P59) specify agile software development.

4.3. RQ2—What SEC models or frameworks have been developed?

In total, 14 different SEC models or frameworks were identified from the primary studies. A comprehensive study on SEC without the foundations of this area—that is, what is required to understand SEC—would mean an incomplete job. The models or frameworks enable the software community to understand the underlying logic of SEC regarding software development while enabling further SEC research to build on them. Table 11 presents the details of the models and frameworks found in the primary studies and their usage as stated in the study, presented according to the research areas identified in Section 4.2.

Five of the models or frameworks are in the area of personnel competence relating to the competence of individuals, overweighed by nine models or frameworks presented for the organizational competence area. In the area of personnel competence research, three of the models or frameworks are for identifying the competencies of software professionals generically (P01, P03, P07), one is for defining the roles and competencies of software testers (P08), and one is for assessing the competencies of software professionals (P14). The models or frameworks in the organizational research area are for managing competence research and learning (P5, P13), a competence process model—for the design, development, and implementation of software (P12, P15), human resource assignment (P10), competence evolution (P04, P09), and competence stakeholder identification (P02, P06).

Table 7

Overview of the primary studies published in journals.

No.	Title of journal	No. of studies	Primary studies
1	Journal of Systems and Software	3	P22, P32, P49
2	The Institution of Engineering and Technology Software	2	P08, P12
3	Information and Software Technology	2	P10, P59
4	IEEE Transactions on Education	1	P28
5	Computers in Human Behavior	1	P33
6	International Journal of Human Capital and Information Technology Professionals	1	P37
7	International Journal of Information Management	1	P50
8	International Journal of Project Management	1	P09
9	Journal of Educational Computing Research	1	P44
10	Journal of Global Information Technology Management	1	P46
11	Economic and Industrial Democracy	1	P34
12	Scientia Iranica	1	P31

Table 8

Research methods of the primary studies.

Research method	Number (percentage)	Primary studies
Mixed-method	28 (46.7%)	P01, P02, P06, P07, P17, P18, P20, P22, P25, P26, P27, P29, P30, P32, P37, P39, P41, P42, P46, P48, P49, P52, P53, P56, P57, P58, P59, P60
Qualitative	21 (35%)	P03, P04, P05, P08, P09, P12, P13, P14, P15, P16, P19, P21, P24, P34, P35, P38, P43, P45, P47, P51, P54
Quantitative	11 (18.3%)	P10, P11, P23, P28, P31, P33, P36, P40, P44, P50, P55

Table 9

Main contributions of the primary studies* (adapted from (Shaw, 2003; Theisen et al., 2017).

Contribution type	Total no.	Primary studies
Frameworks, methods, techniques	6	P01, P02, P07, P09, P13, P14
Guidelines	3	P28, P42, P54
Lessons learned	42	P11, P16, P17, P18, P19, P20, P21, P22, P23, P24, P25, P26, P27, P29, P30, P31, P32, P33, P34, P35, P36, P37, P38, P39, P40, P41, P43, P44, P45, P46, P47, P48, P49, P50, P51, P52, P55, P56, P57 P58, P59, P60
Model	8	P03, P04, P05, P06, P08, P10, P12, P15
Tools	1	P53

* The contribution types are what were specified by the authors of the primary studies.
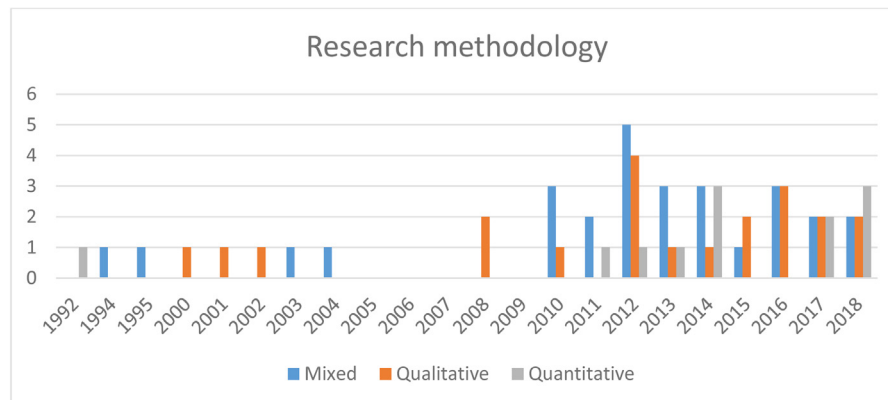
Fig. 2. Overview of the research methodologies used in the primary studies.

4.4. RQ3—What are the essential competencies of software professionals?

As defined by Turley and Bieman (1995), essential competencies of software engineering are the skills, knowledge, and attitudes of software professionals necessary for excellent performance in a software project. Thus, in RQ3, we sought to identify those competencies that induced excellent performance in a software development project. That is competencies defined in the literature as the "essential competencies", "most relevant competencies", "most needed competencies", and "most valued competencies" for software development or similar expressions. The themes that emerged from the identified essential competencies in the studies through the mapping study were programming/coding skills, problem analyzer skills, project management skills, teamwork skills, leadership skills, communication skills, customer-oriented skills, and creative skills. From our primary studies, P16, P17, P18, P19, P20, P21, P22, P23, and P24 are studies that discuss essential competencies.

Table 12a shows 49 essential competencies identified in nine primary studies. The coded items identified in the primary studies were classified and 11 themes or meanings emerged from those essential competencies. The themes were mapped to the top-level

Table 10

The main research areas of the primary studies.

Main area	Subcategory	Description	Primary studies	No. of studies
Personnel competence	Soft competence	Research that looks purely at soft/behavioral competencies of software professionals	P16, P19, P39, P52, P56, P60	6
	Hard competence	Research that looks purely at hard/technical competencies of software professionals	P37, P38, P40, P41, P42, P43	6
	Both soft & hard competencies	Research that examines soft/behavioral and hard/technical competencies of software professionals	P17, P18, P20, P21, P22, P23, P24, P31, P32, P33, P34, P35, P36, P44, P45, P50, P57, P59	18
Organizational competence	Competence identification	Research that investigates tools/instruments for identifying and developing developers' competencies	P01, P02, P07, P11, P53, P54, P58	7
	Competence assessment/measurement	Research that investigates tools/instruments for assessing and measuring developers' competencies	P03, P04, P05, P06, P08, P09, P10, P12, P13, P14, P15, P27, P28, P29, P30	15
	Competence management	Research that combines all the research areas listed above	P25, P26, P46, P47, P48, P49, P51, P55	8
TOTAL			60	60

Table 11

SEC models or frameworks proposed by the primary studies.

Research areas	Primary studies	Model/framework name	Usage as stated in the primary study
Personnel competence	P01	Framework for software development	Identifying soft competencies of software engineers
	P03	SECAT – A Software Engineering Competence Assessment Tool	For assessing software engineering competencies
	P07	Competence framework for software engineers	Identifying a set of knowledge, abilities, and behavior of software engineers
	P08	Competence model for testing teams	Defining the roles and competencies of software testers
	P14	Competence Learning framework (CoLeaF)	Identifying competence expected from graduates
Organizational competence	P15	Process Assessment Model (PAM)	Assessment model for technology and business competence for software development
	P02	SPI competence framework	Identification of stakeholders of software process improvement
	P04	Relationship-based competence management	Collaboration of competence resources
	P05	MoKoM competence model	Informatics competence model and measuring instrument for computer education
	P06	Integrated competence model for entrepreneurs in software	Entrepreneurship development in the software industry
	P09	Competence evolution framework	The development of software competence over time for contractual software development
	P10	Model for assigning human resources	Assigning human resources to a software project
	P12	Process reference model (PRM) and assessment model	Assessing leadership and teams in global software development
	P13	Framework for research and learning process	Research and learning process for defining competence in teamwork in computer education

themes of Rivera-Ibarra et al.'s (2010) framework for identifying competencies. Out of the 49 essential competencies identified from the literature, 18 represented technical knowledge or skills. These are referred to as the essential hard competencies and are defined as task-oriented competencies. Also, 10 out of the 49 represented interpersonal skills. These are competencies for organizing cooperation and interpersonal relations in a software development project. Additionally, 21 out of the 49 essential competencies were personal knowledge or skills, which are personal attributes for working well in different spheres of life. Both social and personal knowledge or skills count in the area of soft competencies. Therefore, there appears to be a larger group of essential soft competencies than essential hard competencies.

In Table 12b, we present the technical or hard essential competencies reported by primary studies and map them to their associated software engineering roles, which are presented in Table 1. As such, Table 12b demonstrates the hard competencies considered essential for holding a software engineering role and performing its associated tasks.

Although Table 12b shows no essential technical competence for the role of configuration and maintenance engineer, this does not presuppose that we lack essential technical competence for that role. Given this, we call for using broader empirical data to extensively explore all the roles. The roles used in Table 12b are from SWEBOK. As stated in Section 2.1, the role can be associated with the position and different tasks. However, in this table, we

Table 12a

Essential SEC as reported by the primary studies.

Codes (essentials from the literature)	Primary studies	Theme name for the essential competence and description	Type of competence (top-level theme of Rivera-Ibarra et al., 2010)
Writes/automates tests with code, leverages/reuses code, uses code reading, uses structured techniques for communication	P22, P18, P21	**Programming/coding skills** —essential hard skills for performing the coding function in software development	Technical knowledge
Skills/techniques for architecture design, use of prototypes to assess design, possesses unique domain knowledge, obtains necessary training/learning, experience with a similar project, technical know-how, uses methodical problem solving, use of new methods and tools, thinking—strong analytic skills, using techniques and tools, methodological competence, analytical thinking	P22, P20, P18, P17	**Problem analyzer skills**—essential hard skills for performing requirements and problem analyzing in software development	Technical knowledge
Schedules and estimates well, responds to schedule pressure by sacrificing parts of the design process, organizational skills	P22, P18, P20	**Project management skills**—essential skills for managing schedules in software developments	Technical knowledge
Team oriented, interpersonal relationship	P22, P18, P20	**Teamwork skills**—essential soft social skills needed for performing software developments in teams	Interpersonal skills
Proactive role with management, helps others, information sharing, willingness to confront others, scalability	P22, P18, P20	**Leadership skills**—essential soft or social skills for performing a leadership role in software development	Interpersonal skills
Good written communication, good verbal communication, reflection	P20	**Communication skills**—an essential soft social skill for performing communications in software developments	Interpersonal skills
Focus on users or customer needs	P22, P18	**Customer-oriented skills**—essential soft social skills for performing customer relationships in software development	Interpersonal skills
Creative thinking skills, being innovative, attention to detail, desire to improve things, design style	P19	**Creative skills**—essential soft personal skills for innovation creation in software development	Personality traits
Perseverance, the strength of convictions, proactive/initiator/driver, lack of ego, driven by a desire to contribute, desire to do/bias for action, sense of fun, driven by a sense of mission, mixes personal and work goals, thoroughness, seeks help	P22, P18, P24, P16	**Self-esteem skills**—essential soft personal attribute of an individual for evaluating their own worth in software development	Personality traits
Maintaining a "big picture" view, breadth of view and influence, emphasizes elegant and simple solutions	P22, P18	**Essence skills**—an essential soft personal attribute of an individual for performing abstraction in software development	Personality traits
Quality, pride in quality, and productivity	P22, P18	**Quality control skills**—quality control skills, essential	Personality traits

Table 12b

Essential technical competence mapped with software professionals roles.

Roles	Essential technical competencies reported by the primary studies	Primary studies
Requirement analyst	Schedules and estimates well (estimating requirement workload), organizational skills	P22, P18, P22
Designer	Skills/techniques for architecture design, use of prototypes to assess the design, response to schedule pressure by sacrificing parts of the design process, possesses unique domain knowledge, obtains necessary training/learning, experience with a similar project, technical know-how, uses methodical problem solving, use of new methods and tools, thinking—strong analytic skills, using techniques and tools, methodological competence, analytical thinking	P22, P20, P18, P17
Programmer	Uses structured techniques for communication, leverages/reuses code, uses code reading	P18, P21, P22
Software test & quality engineer	Writes/automates tests with code	P18, P22
Configuration & maintenance engineer	–	–

map these essential competencies to specific roles to indicate how these tasks and roles can be matched.

4.5. RQ4—How has SEC research evolved over the last three decades?

This research question aims to identify changes in SEC research over the past two and a half decades. Knowing these changes may guide further research on how to tackle emerging issues in the SEC research area. The first trend that we observed from the primary studies is an increase in the total number of primary studies that used quantitative research methods since 2011.

Regarding the contribution types, it seems that proposing a set of lessons learned continues to be the focus of SEC research, as the number of primary studies with this type of contribution remained the highest over time. Regarding research areas, however,
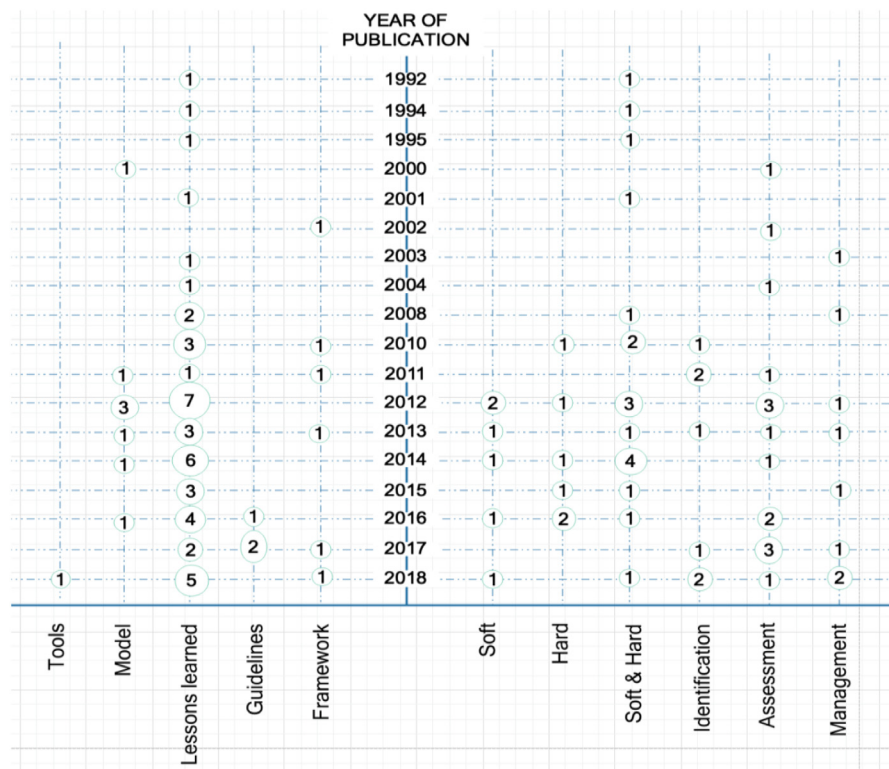
YEAR OF PUBLICATION

Year	Tools	Model	Lessons learned	Guidelines	Framework	Soft	Hard	Soft & Hard	Identification	Assessment	Management
1992			1			1					
1994			1			1					
1995			1			1					
2000		1								1	
2001			1			1					
2002					1					1	
2003											1
2004			1								1
2008			2					1			1
2010			3				1	2	1		
2011		1	1						2	1	
2012		3	7			2	1	3		3	1
2013		1	3	1		1	1			1	1
2014		1	6			1	1	4		1	
2015			3				1	1			1
2016		1	4	1		1	2	1		2	
2017			2	2	1				1	3	1
2018	1		5		1	1		1	2	1	2

Fig. 3. The evolution of SEC research sub-areas (right) and the contribution of the primary studies (left) over time.

it seems that since 2012, soft competencies have been receiving some attention from SEC research. Fig. 3 shows that, since 2012, there have been six studies (P16, P19, P39, P52, P56, and P60) focusing on soft skills and 11 studies (P17, P20, P21, P23, P31, P32, P33, P35, P36, P44, and P57) focusing on both soft and hard competencies.

Another trend observed in essential competence studies is the number of competencies identified per study. An early study by Turley and Bieman (1995) identified 38 essential competencies. In the period from 1996 to 2009, no studies emerged on this topic, and the subsequent publications date to 2010, 2012, 2016, 2017, and 2018. Four studies (P16, P17, P19, and P24) considered soft competencies, each focusing on only a single essential competence. Study P20 considered both soft and hard competencies. Lastly, study P21 identified two hard competencies. We can summarize that except for the 1995 paper by Turley and Bieman (1995), that is, P22, the later studies dating in the 2010s have been considering fewer essential competencies. In fact, the highest number of essential competencies identified in one of the later studies was only four. Again, most of the studies focused on soft competencies. All the mentioned papers are conference publications. The publication of initial research findings at a conference, with a narrower focus, could explain why the number of essential competencies mentioned in them is rather low. Thus, it has been more than two decades since the introduction of the topic and the initial identification of the essential competencies. This highlights the need for a fresh understanding of the essential competencies that particularly considers current software development trends, such as using agile methods (Abrahamsson et al., 2002) and DevOps (Debois, 2011).

5. Discussion

By conducting an extensive mapping study, we reviewed the extant literature on SEC. Similar to previous review studies (Cruz et al., 2015), our findings show that the number of publications on SEC has been increasing in recent years. This indicates that although SEC is a mature field of study (Lenberg et al., 2015), it continues to be a strategic research area for the software engineering discipline (Colomo-Palacios et al., 2013). Regarding the use of research methods in primary studies, we observed that quantitative methods have gained some prominence in SEC studies in recent years. This could be another indication of the maturity of the SEC research field, as the application of quantitative methods requires a consensus of theories and concepts (Palvia et al., 2004). Regarding their contributions, our results show that most primary studies provide a set of lessons learned. Specifically, among the primary studies published before 2011, almost all reported a set of lessons learned as their main contribution. This observation could indicate a widespread utilization of SEC knowledge in practice. However, it also highlights the lack of relevant frameworks necessary for sustaining a consistent body of knowledge on SEC, and facilitating the construction and management of software (Shaw, 2003; Theisen et al., 2017).

5.1. Summary of findings

Our study has several implications for SEC research and practice. While previous review studies (Cruz et al., 2015; Moustroufas et al., 2015; Vishnubhotla et al., 2018) have focused only on specific areas of SEC, we provided a broader multi-perspective overview of the state of SEC research. As such, we synthesized the previous literature on SEC from different angles. First, our results show that the identified primary studies focus on the two main areas of personnel and organizational competencies. The first area focuses on the soft and hard competencies of individual software professionals that are necessary for developing software. This observation agrees with previous review studies (Cruz et al., 2015; Stepanenko and Kashevnik, 2017; Vishnubhotla et al., 2018), which have reported individual competencies as

the main topic of interest for software engineering researchers. The second area is mainly concerned with the instruments necessary for identifying, evaluating, and managing the personnel competencies for successful software development.

As mentioned by Mead and Shoemaker (2013), the SEC models and frameworks are beneficial in organizing and observing the understanding of SEC. Therefore, we set out to identify the instruments reported by the primary studies. In total, 14 models or frameworks for observing and understanding SEC were proposed by the primary studies. As claimed by the primary studies, these models and frameworks can be used for three main purposes: first, identifying competencies of software professionals (Espinosa-curiel et al., 2011; Orsoni and Colaco, 2013; Papoutsoglou et al., 2018; Rivera-Ibarra et al., 2010; Saldaña-Ramos et al., 2012; Seppanen, 2000); second, assessing the competencies of software professionals (Salviano et al., 2012; Sedelmaier and Landes, 2014a; Tuffley, 2012); and third, developing SEC (Frezza et al., 2018; Linck et al., 2013; Pérez et al., 2017; Schulte et al., 2017). According to Frezza et al. (2018) and IEEE (2014), competence development is not the sole responsibility of software professionals. It also requires the participation of other stakeholders, including software companies and the software industry in general, alongside educators who provide education and training to software professionals. Therefore, we claim that the identification, assessment, and development of SEC are interrelated functions, which must be considered as a whole and from the perspective of these three groups of stakeholders. However, our results show a lack of comprehensive models or frameworks that could be used for this purpose. Additionally, we did not find any model or framework for assessing the benefit of various SEC for these stakeholders, such as determining satisfaction levels derived from possessing or using a competence. Therefore, future studies are needed to propose better models or frameworks for providing theoretical accounts along with practical implications for different aspects of SEC, especially for assessing the SEC stakeholders' satisfaction levels derived from possessing or using a certain competence or set of competencies.

Another goal of our study was to identify the essential SEC that are considered necessary for exceptional performance in a software project (Turley and Bieman, 1995). To that end, we identified 49 unique essential competencies reported by the primary studies and classified them into 11 themes. Of these 49 essential competencies, 18 were classified as hard and 31 as soft competencies. Since hard competencies are considered necessary to perform a job (Urs, 2013), they can be mapped to different software engineering roles. Therefore, we mapped the essential hard competencies to different roles. Surprisingly, however, we did not find any essential hard competencies for the role of configuration and maintenance engineer because, traditionally, this role has been seen as a part of the operations function rather than software development. However, considering recent shifts toward the application of DevOps (Debois, 2011), this gap becomes concerning. Therefore, future empirical research is needed to provide a better understanding of how the essential competencies of software professionals change over time or when they move from one role to another.

Finally, our results show that in recent years a higher number of primary studies have focused on studying soft competencies, and concurrently, a higher number of essential soft competencies have been reported. This observation could be because the shift toward using agile methods (Feldt et al., 2010; Licorish and Macdonell, 2013; Papoutsoglou et al., 2018)—which generally emphasize the role of individuals, their personality traits, and attitudes—is more related to soft skills. However, since 12 more recent primary studies focus on competencies specifically regarding agile development, it seems that this issue has started

getting attention. The small number of primary studies focusing on SEC regarding agile development may also indicate a slight delay in incorporating agile methods into the software engineering education curricula (Moreno et al., 2012). Such a delay would be no surprise, as educational institutions, in planning their curricula and defining the required learning outcomes for their degrees, are generally very considerate in their response to the employment market demand (Ghezzi and Mandrioli, 2005) and base the curriculum changes on fundamental issues supported by research. Today, we can expect that the shift is already occurring at educational institutions as well (Schulte et al., 2017). Nevertheless, further research is needed to assess and provide a fresh account of the essential competencies of future software professionals, especially regarding modern development methods and techniques.

Another observation is that, going over to agile software development, the focus in competence development is no longer solely within the organization and project teams developing software, but importantly, this also sets requirements for the customers (Kropp et al., 2016; Robal et al., 2015). Agile software development, in essence, involves intense communication and collaboration with the customer throughout the development life cycle. The counterpart on the customer side should also bring into the project some essential competencies to transfer domain knowledge and support in defining requirements or prioritizing these. If this is not happening, the efforts to communicate and interact on the part of the software professionals alone cannot ensure the success of the software project. This point might be neglected in software development generally, but in the agile approach, this becomes a key issue.

5.2. Threats to validity

The common threats that affect mapping studies are biases in the selection process and inaccuracies in data extraction. This paper is no exception, and our results could have been influenced by these limitations. Below, we discuss the threats to the validity of our results based on the framework suggested by Wohlin et al. (2012) and mention the steps taken to mitigate them.

Regarding construct validity, there is a concern about using the right measure for studying SEC. To avoid potential misconceptions, the definitions and keywords were carefully selected from previous SEC studies and with the contribution of all members of the research team. Additionally, to identify relevant literature, the identified keywords were used to develop and test a search string. Although we used known papers from some selected literature reviews in the SEC area to validate the use of the term "competence" in the string statement (as against terms such as skills, knowledge, attitude, etc.), we acknowledge that we may have missed some papers during the selection process. Nevertheless, a thoughtful process was considered before the use of the term competence. To avoid personal mistakes and misinterpretations, although the first author took the lead in implementing the protocol, each step of the iterative review process was evaluated and verified by other authors. For instance, the inclusion and exclusion criteria were applied by the first author, while in each round, the second author randomly selected some studies and verified the decisions made by the first author. Where there was disagreement between the authors, all the authors engaged in face-to-face discussions to reach a consensus about including or excluding a study.

Concerning external validity, our results are based on an extensive review of previous literature. However, there could have been limitations in the selection of primary studies. Considering the multidisciplinary nature of the topic, we tried to search major databases that cover the software engineering literature

Table A.1

No.	Test string	Number of items retrieved
1	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND ("research* area*" OR framework* OR essential* OR evol*)*	8,820
2	*(software AND develop* OR system* AND develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND ("research*area*" OR framework* OR essential* OR evol*) AND (LIMIT-TO (SUBAREA, "COMP") OR LIMIT-TO (SUBAREA, "ENGI"))*	6,580
3	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND ("research* area*" OR "research* type*" OR scope OR anatom)*	648
4	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND (framework* OR model* OR theor* OR structur*)*	14,975
5	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND (essential* OR important* OR valu*)*	6,870
6	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND (evol* OR time)*	8,680
7	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND (("research* area*" OR "research* type*" OR scope OR anatom*) AND (framework* OR model* OR theor* OR structur*) AND (essential* OR important* OR valu*) AND (evol* OR time))*	340
8	*(software develop* OR system* develop) AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*)*	149
9	("software develop*" OR "system* develop*") AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*)	45,054
10	("software develop*" OR "system* develop*") AND (competenc* OR knowledge* OR skill* OR abilit* OR attitud*) AND (LIMIT-TO (SUBAREA,"COMP") OR LIMIT-TO (SUBAREA,"ENGI"))	29,112

and its related areas, such as information systems and computer science. Therefore, we decided to limit the scope of our search to IEEE Xplore, ACM Digital Library, Scopus, AIS eLibrary, and ScienceDirect databases, as they cover most software engineering journals and conferences. However, the authors acknowledge that the results of the mapping study are limited to only scientific studies that are accessible through the selected databases. As we discussed in Section 3.3, although nine out of the 11 criteria proposed by Dyba and Dingsoyr (2008) could be used to assess the quality of both empirical and conceptual studies, the use of criteria could have led to a bias in selecting empirical studies. To mitigate this threat, we decided to choose the exclusion threshold (i.e., four) based on the number of criteria that apply to both empirical and conceptual studies.

Internal validity, which is related to causal relationships of SEC, was not the aim of the study. Hence, a statistical inference was not applied in this study for causal relationships. Therefore, internal validity as a threat did not affect this study.

Finally, conclusion validity is related to bias on the part of the researchers in interpreting the data. It is important to note that this threat cannot be totally eliminated; however, measures were taken to mitigate it. First, we developed and tested a predefined research protocol based on the well-established guidelines proposed by previous research (Kitchenham and Charters, 2007; Petersen et al., 2008, 2015). Three authors were involved in developing, testing, and improving the protocol. Next, although data extraction was performed by the first author, the second author verified the consistency and validity of the extracted data before the data analysis was conducted by all the authors. Additionally, all the authors were involved in the data analysis and synthesis of the results. Finally, the results of the study were discussed and reviewed by the whole research team to increase the reliability of the results.

6. Conclusions

In this paper, we aimed to give a comprehensive overview of the current state of research on SEC. To achieve this goal, we conducted an extensive systematic mapping study and found 60 relevant primary studies for closer examination. By analyzing these studies, we identified two main SEC research areas: the personnel and organizational areas. Additionally, we identified and presented several SEC models and frameworks produced in

prior research that could be used by SEC research and practice. Furthermore, we identified a set of competencies of software professionals that are considered essential. Most of them deal with social and personal skills and competencies (i.e., the so-called soft skills). Based on this observation, we argue that separating soft and hard competencies may soon be a thing of the past, and future research and practice should consider them as two equally critical pillars of SEC. Our results show that, among other things, the human-intensive nature of software development requires further attention from both research and practice. Therefore, we argue that developing SEC cannot be conducted in isolation but must consider the viewpoints of different SEC stakeholders, including software professionals, educators, and the software industry. Finally, future research should identify and provide a better understanding of the essential SEC, contributing to developing high-quality software products and systems in modern societies.

CRediT authorship contribution statement

Nana Assyne: Conceptualization, Methodology, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Hadi Ghanbari:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision. **Mirja Pulkkinen:** Methodology, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Appendix A. Different search strings tested to develop the search string used in this study

See Table A.1. Different search strings tested to develop the search string used in this study.

Table B.1

P01	Orsoni, A., & Colaco, B. (2013). A competency framework for software development organisations. *2013 UKSim 15th International Conference on Computer Modelling and Simulation*, 507–511.
P02	Espinosa-Curiel, I. E., Rodríguez-Jacobo, J., & Fernández-Zepeda, J. A. (2011). A competency framework for the stakeholders of a software process improvement initiative. *International Conference on Software and Systems Process*, 139–148.
P03	Sedelmaier, Y., & Landes, D. (2014). A multi-perspective framework for evaluating software engineering education by assessing students' competencies: SECAT—A software engineering competency assessment tool. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–8.
P04	Seppanen, V. (2000). A relationship-based view to software engineering competence. *International Conference on Product Focused Software Process Improvement*, 376–390.
P05	Linck, B., Ohrndorf, L., Schubert, S., Stechert, P., Magenheim, J., Nelles, W., Neugebauer, J., & Schaper, N. (2013). Competence model for informatics modelling and system comprehension. *2013 IEEE Global Engineering Education Conference (EDUCON)*, 85–93.
P06	Aisha, A. N., Siswanto, J., & Sudirman, I. (2016). Competencies model for entrepreneur development in software industries. *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 184–188.
P07	Rivera-Ibarra, J. G., Rodríguez-Jacobo, J., Fernández-Zepeda, J. A., & Serrano-Vargas, M. A. (2010). Competency framework for software engineers. *2010 23rd IEEE Conference on Software Engineering Education and Training*, 33–40.
P08	Saldaña-Ramos, J., Sanz-Esteban, A., García-Guzmán, J., & Amescua, A. (2012). Design of a competence model for testing teams. *IET Software*, 6(5), 405–415.
P09	Seppänen, V. (2002). Evolution of competence in software subcontracting projects. *International Journal of Project Management*, 20(2), 155–164.
P10	André, M., Baldoquín, M. G., & Acuña, S. T. (2011). Formal model for assigning human resources to teams in software projects. *Information and Software Technology*, 53, 259–275.
P11	Papoutsoglou, M., Kapitsaki, G. M., & Mittas, N. (2018). Linking personality traits and interpersonal skills to gamification awards. *2018 44th Euromicro Conference on Software Engineering and Advanced Applications*, 214–221.
P12	Tuffley, D. (2012). Optimising virtual team leadership in global software development. *IET Software*, 6(3), 176–184.
P13	Schulte, C., Magenheim, J., Muller, K., & Budde, L. (2017). The design and exploration cycle as research and development framework in computing education. *2017 IEEE Global Engineering Education Conference (EDUCON)*, April, 867–876.
P14	Frezza, S., Daniels, M., Pears, A., Cajander, A., Viggo, K., Kapoor, A., McDermott, R., Peters, A.-K., Sabin, M., & Wallace, C. (2018). Modelling competencies for computing education beyond 2020: A research based approach to defining competencies in the computing disciplines. *23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 148–174.
P15	Salviano, C. F., Alves, A. M., Stefanuto, G. N., Maintinguer, S. T., Mattos, C. V, Zeitoum, C., & Reuss, G. (2012). Developing a process assessment model for technological and business competencies on software development. *2012 Eighth International Conference on the Quality of Information and Communications Technology*, 15504, 125–130.
P16	Engelbrecht, L., Landes, D., & Sedelmaier, Y. (2018). A didactical concept for supporting reflection in software engineering education. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 547–554.
P17	Sedelmaier, Y., & Landes, D. (2012). A research agenda for identifying and developing required competencies in software engineering. *2012 15th International Conference on Interactive Collaborative Learning (ICL)*, 01, 1–5.
P18	Turley, R. T., & Bieman, J. M. (1994). Identifying essential competencies of software engineers. *Proceedings of the 22nd Annual ACM Computer Science Conference on Scaling Up: Meeting the Challenge of Complexity in Real-World Computing Applications, CSC '94*, 271–278.
P19	Chang, J. W., Wang, T. I., Lee, M. C., Su, C. Y., & Chang, P. C. (2016). Impacts of using creative thinking skills and open data on programming design in a computer-supported collaborative learning environment. *Proceedings—IEEE 16th International Conference on Advanced Learning Technologies, ICALT 2016*, 396–400.
P20	Calazans, A., Paldês, R., Masson, E., Rezende, K., Braosi, E., Perera, N., & Brito, I. S. (2017). Software requirements analyst profile: A descriptive study of Brazil and Mexico. *2017 IEEE 25th International Requirements Engineering Conference*, 204–212.
P21	Suhartono, J., & Sudirwan, J. (2016). Academic competence of computer science graduate degree from the employer's perspective. *2016 International Conference on Information Management and Technology (ICIMTech)*, November, 176–181.
P22	Turley, T., & Bieman, M. (1995). Competencies of exceptional and non-exceptional software engineers. *Journal of Systems Software*, 28(1), 19–38.
P23	Gold, C., Abke, J., & Sedelmaier, Y. (2014). A retrospective course survey of graduates to analyse competencies in software engineering. *2014 IEEE Global Engineering Education Conference (EDUCON)*, April, 100–106.
P24	Magenheim, J., Nelles, W., Rhode, T., Schaper, N., & Schubert, S. (2010). Competencies for informatics systems and modeling results of qualitative content analysis of expert interviews. *IEEE EDUCON 2010 Conference*, 513–521.
P25	Román, S. E., & Mijes, E. L. (2003). Evolution and trends among computer science and computer engineering professionals in Mexico (1971–2005). *33rd Annual Frontiers in Education, 2003. FIE*, 13–17.
P26	Sedelmaier, Y., & Landes, D. (2017a). How can we find out what makes a good requirements engineer in the age of digitalization? *2017 IEEE Global Engineering Education Conference (EDUCON)*, April, 230–238.
P27	Juárez-Ramírez, R., Jiménez, S., & Huertas, C. (2016). Developing software engineering competences in undergraduate students: A project-based learning approach in academy-industry collaboration. *2016 4th International Conference in Software Engineering Research and Innovation*, 87–96.
P28	Pérez, J., Vizcarro, C., García, J., Bermúdez, A., & Cobos, R. (2017). Development of procedures to assess problem-solving competence in computing engineering. *IEEE Transactions on Education*, 60(1), 22–28.
P29	McNamara, R. A. (2004). Evaluating assessment with competency mapping. *Sixth Australasian Conference on Computing Education*, 193–199.
P30	Juárez-Ramírez, R., & Jiménez, S. (2017). Towards assessing attitudes and values in the practice of software engineering: The competency-based learning approach. *2017 5th International Conference in Software Engineering Research and Innovation*, 153–162.

Table B.1 (*continued*).

P31	Alavi, S. B., Moteabbed, S., & Arasti, M. R. (2012). A qualitative investigation of career orientations of a sample of Iranian software engineers. *Scientia Iranica*, *19*(3), 662–673.
P32	Moreno, A. M., Sanchez-Segura, M. I., Medina-Dominguez, F., & Carvajal, L. (2012). Balancing software engineering education and industrial needs. *Journal of Systems and Software*, *85*(7), 1607–1620.
P33	Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F. J., & Tovar-Caro, E. (2013). Competence gaps in software personnel: A multi-organisational study. *Computers in Human Behaviour*, *29*(2), 456–461.
P34	Marks, A., & Scholarios, D. (2008). Choreographing a system: Skill and employability in software work. *Economic and Industrial Democracy*, *29*(1), 96–124.
P35	Zehetmeier, D., Böttcher, A., Bruggemann-Klein, A., Thurner, V., & Schlierkamp, K. (2015). Finding competence characteristics among first semester students in computer science. *IEEE Frontiers in Education Conference*, 1–9.
P36	Hiranrat, C., & Harncharnchai, A. (2018). Using text mining to discover skills demanded in software development jobs in Thailand. *ACM International Conference Proceeding Series*, 112–116.
P37	Colomo-Palacios, R., Tovar-Carlos, E., Garcia-Crespo, Å., & Gómez-Berbís, J. M. (2010). The case of software engineers identifying technical competences of IT professionals. *International Journal of Human Capital and Information Technology Professionals*, *1*(March), 31–43.
P38	Köppe, C. (2012). Using pattern mining for competency-focused education. *2nd Computer Science Education Research Conference*.
P39	Sedelmaier, Y., & Landes, D. (2014b). Software Engineering Body of Skills (SWEBOS). *IEEE Global Engineering Education Conference, EDUCON*, *April*, 395–401.
P40	Masrek, M. N., Osman, M. A. F., Ibrahim, Z., & Mansor, A. N. (2015). Malaysian computer professional: Assessment of emotional intelligence and organisational commitment. *Procedia – Social and Behavioural Sciences*, *172*, 238–245.
P41	Robal, T., Ojastu, D., Kalja, A., & Jaakkola, H. (2015). Managing software engineering competences with domain ontology for customer and team profiling and training. *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, 1369–1376.
P42	Samuelsen, T., Colomo-Palacios, R., & Kristiansen, M. (2016). Learning software project management in teams with diverse backgrounds. *Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality – TEEM 16*, 127–131.
P43	Kropp, M., Meier, A., & Perellano, G. (2016). Experience report of teaching agile collaboration and values agile software development in large student teams. *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, 76–80.
P44	Zendler, A., Klaudt, D., & Seitz, C. (2014). Empirical determination of competence areas to computer science education. *Journal of Educational Computing Research*, *51*(1), 71–89.
P45	Iivari, J., Hirschheim, R., & Klein, H. (2001). Towards more professional information systems development: ISD as knowledge work. *ECIS 2001 Proceedings*, *35*, 1025–1036.
P46	Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J. (2012). Analyzing human resource management practices within the GSD context. *Journal of Green Information Technology*, 1–27.
P47	Romero, M., Vizcaíno, A., & Piattini, M. (2008). Competencies desirable for a requirements elicitation specialist in a global software development. *ICSOFT*, *January*, 347–354.
P48	Paasivaara, M., Lassenius, C., Damian, D., Petteri, R., & Schröter, A. (2013). Teaching students global software engineering skills using distributed scrum. *2013 35th International Conference on Software Engineering (ICSE)*, 1128–1137.
P49	Fagerholm, F., Hellas, A., Luukkainen, M., Kyllönen, K., Yaman, S., & Mˊaenpˋaˊa, H. (2018). Designing and implementing an environment for software start-up education: Patterns and anti-patterns. *The Journal of Systems & Software*, *146*, 1–13.
P50	Broadbent, M., Lloyd, P., Hansell, A., & Dampney, C. N. G. (1992). Roles, responsibilities and requirements for managing information systems in the 1990s. *International Journal of Information Management*, *72*(12), 21–38.
P51	Seppänen, P., Liukkunen, K., & Oivo, M. (2015). On the feasibility of startup models as a framework for research on competence needs in software startups. *International Conference on Product-Focused Software Process Improvement*, *9459*, 569–576.
P52	Licorish, S. A., & MacDonell, S. G. (2013). Differences in jazz project leaders' competencies and behaviours: A preliminary empirical investigation. *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering*, 1–8.
P53	Calikli, G., Al-Eryani, M., Baldebo, E., Horkofff, J., & Ask, A. (2018). Effects of automated competency evaluation on software engineers' emotions and motivation: A case study. *Proceedings – International Conference on Software Engineering*, 44–50.
P54	Sedelmaier, Y., & Landes, D. (2017a). Experiences in teaching and learning requirements engineering on a sound didactical basis. *ITiCSE*, 116–121.
P55	Wang, C., Cui, P., Daneva, M., & Kassab, M. (2018). Understanding what industry wants from requirements engineers: An exploration of RE jobs in Canada. In ACM (Ed.), *Proceedings of the 12th ACM/IEEE Empirical Software Engineering and Measurement* (pp. 1–10).
P56	Thurner, V., & Böttcher, A. (2012). Expectations and deficiencies in soft skills evaluating student competencies in software engineering education. *IEEE Global Engineering Education Conference (EDUCON)*.
P57	Thurner, V., Axel, B., & Andreas, K. (2014). Identifying base competencies as prerequisites for software engineering education. *IEEE Global Engineering Education Conference (EDUCON)*, *April*, 1069–1076.
P58	Duke, D., & Hribar, L. (2011). Implementation of software competence development process. *2011 Proceedings of the 34th International Convention MIPRO – IEEE*, 587–592.
P59	Feldt, R., Angelis, L., Torkar, R., & Samuelsson, M. (2010). Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, *52*(6), 611–624.
P60	Patel, A., Benslimane, Y., Bahli, B., & Yang, Z. (2012). Addressing IT security in practice: Key responsibilities, competencies and implications on related bodies of knowledge. *2012 IEEE International Conference on Industrial Engineering and Engineering Management*, 899–903.

Appendix B. List of the primary studies

See Table B.1. List of the primary studies.

Appendix C. List of the conferences targeted by the primary studies

See Table C.1. List of the conferences targeted by the primary studies.

Table C.1

No.	Channel title of conference	No. of publications	Primary studies
1	IEEE Global Engineering Education Conference (EDUCON)	9	P56, P05, P23, P39, P57, P13, P26, P16, P24
2	Frontiers in Education	3	P25, P35, P03
3	ACM Conference on Innovation and Technology in Computer Science Education	2	P54, P14
4	IEEE International Conference on Industrial Engineering and Engineering Management	2	P60, P06
5	International Conference on Product-Focused Software Process Improvement	2	P04, P51
6	International Conference in Software Engineering Research and Innovation	2	P27, P30
7	Euromicro Conference on Software Engineering and Advanced Applications (SEAA)	1	P11
8	ACM Conference on Computer Science	1	P18
9	Global Conference on Business & Social Science	1	P40
10	International Conference on Software Technologies	1	P47
11	IEEE Conference on Software Engineering Education and Training	1	P07
12	Australasian Conference on Computing Education	1	P29
13	IEEE International Conference on Advanced Learning Technologies (ICALT)	1	P19
14	IEEE International Conference on Software Engineering Education and Training	1	P43
15	IEEE International Requirements Engineering Conference (RE)	1	P20
16	IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)	1	P53
17	International Conference on Computer Modelling and Simulation	1	P01
18	International Conference on Education and Multimedia Technology	1	P36
19	International Conference on Information Management and Technology	1	P21
20	International Conference on Interactive Collaborative Learning (ICL)	1	P17
21	International Conference on Software and Systems Process	1	P02
22	International Conference on Software Engineering	1	P48
23	International Conference on Technological Ecosystems for Enhancing Multiculturality	1	P42
24	International Conference on the Quality of Information and Communications Technology	1	P15
25	International Convention MIPRO	1	P58
26	International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)	1	P52
27	Portland International Conference on Management of Engineering and Technology	1	P41
28	ACM/IEEE International Symposium on Empirical Software Engineering and Measurement	1	P55
29	Computer Science Education Research Conference	1	P38

References

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., 2002. Agile software development methods: Review and analysis. VTT Publ. 478, 3–107.

Acuña, S.T., Juristo, N., 2004. Assigning people to roles in software projects. Softw. - Pract. Exp. 34 (7), 675–696. http://dx.doi.org/10.1002/spe.586.

Acuña, S.T., Juristo, N., Moreno, A.M., 2006. Emphasizing human capabilities in software development. IEEE Softw. 23 (2), 94–101. http://dx.doi.org/10.1109/MS.2006.47.

Ahmed, F., Capretz, L.F., Bouktif, S., Campbell, P., 2013. Soft skills and software development: A reflection from software industry. Int. J. Inf. Process. Manage. 4 (3), 171–191. http://dx.doi.org/10.4156/ijipm.vol4.issue3.17.

Andrews, J., Higson, H., 2008. Graduate employability, 'Soft Skills' Versus 'Hard' business knowledge: A European study. High. Educ. Eur. 33 (4), 411–422. http://dx.doi.org/10.1080/03797720802522627.

Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I.O., Jaccheri, L., 2018. Software startup engineering: A systematic mapping study. J. Syst. Softw. 144 (June), 255–274. http://dx.doi.org/10.1016/j.jss.2018.06.043.

Bourque, P., Fairley, R.E., 2014. Guide to the Software Engineering - Body of Knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press, http://dx.doi.org/10.1234/12345678.

Bröker, K., 2014. Identification and measurement of computer science competencies in the area of software development, software engineering and programming. In: Proceedings of the Tenth Annual Conference on International Computing Education Research. ICER '14, pp. 141–142. http://dx.doi.org/10.1145/2632320.2632322.

Calazans, A., Paldês, R., Masson, E., Rezende, K., Braosi, E., Perera, N., Brito, I.S., 2017. Software requirements analyst profile : A descriptive study of Brazil and Mexico. In: 2017 IEEE 25th International Requirements Engineering Conference. pp. 204–212. http://dx.doi.org/10.1109/RE.2017.22.

Casale, G., Chesta, C., Deussen, P., Di Nitto, E., Gouvas, P., Koussouris, S., Stankovski, V., Symeonidis, A., Vlassiou, V., Zafeiropoulos, A., Zhao, Z., 2016. Current and future challenges of software engineering for services and applications. Procedia Comput. Sci. 97, 34–42. http://dx.doi.org/10.1016/j.procs.2016.08.278.

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F.J., Tovar-Caro, E., 2013. Competence gaps in software personnel: A multi-organizational study. Comput. Hum. Behav. 29 (2), 456–461. http://dx.doi.org/10.1016/j.chb.2012.04.021.

Cruz, S., Fabio, Q.B., Fernando, L., 2015. Forty years of research on personality in software engineering: A mapping study. Comput. Hum. Behav. 46, 94–113. http://dx.doi.org/10.1016/j.chb.2014.12.008.

Debois, P., 2011. Devops: A software revolution in the making. J. Inf. Technol. Manag. 24 (8), 3–39.

Dyba, T., Dingsoyr, T., 2008. Empirical studies of agile software development: A systematic review. Inf. Softw. Technol. 50, 833–859. http://dx.doi.org/10.1016/j.infsof.2008.01.006.

Espinosa-curiel, I.E., Rodríguez-jacobo, J., Fernández-zepeda, J.A., 2011. A competency framework for the stakeholders of a software process improvement initiative. In: International Conference on Software and Systems Process, pp. 139–148.

Feldt, R., Angelis, L., Torkar, R., Samuelsson, M., 2010. Links between the personalities, views and attitudes of software engineers. Inf. Softw. Technol. 52 (6), 611–624. http://dx.doi.org/10.1016/j.infsof.2010.01.001.

Fink, A., 2010. Conducting Research Literature Reviews: From the Internet to Paper, third ed. SAGE.

Frezza, S., Daniels, M., Pears, A., Cajander, A., Viggo, K., Kapoor, A., Mcdermott, R., Peters, A.-K., Sabin, M., Wallace, C., 2018. Modelling competencies for computing education beyond 2020: A research based approach to defining competencies in the computing disciplines. In: 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, pp. 148–174.

Ghezzi, C., Mandrioli, D., 2005. The challenges of software engineering education. In: International Conference on Software Engineering. pp. 115–127. http://dx.doi.org/10.1007/11949374.

Gimenes, I.M.S., Barroca, L., Barbosa, E.F., 2012. The future of human resources qualifications in software engineering – meeting demands from industry and benefiting from educational and technological advances. In: 2012 26th Brazilian Symposium on Software Engineering. pp. 181–185. http://dx.doi.org/10.1109/SBES.2012.19.

Harzallah, M., Giuseppe, B., Vemadat, F., 2002. A formal model for assessing individual competence in enterprises. IEEE Int. Conf. Syst. Man Cybern. 6, http://dx.doi.org/10.1109/ICSMC.2002.1173300.

Harzallah, M., Vernada, F., 2002. It-based competency modeling and management: from theory to practice in enterprise engineering and operations. Comput. Ind. 48, 157–179.

Havelka, D., Mermout, Jeffrey W., 2009. Toward a theory of iformation technology professional competence. J. Comput. Inf. Syst. Winter.

Holtkamp, P., Jokinen, J.P.P., Pawlowski, J.M., 2015. Soft competency requirements in requirements engineering, software design, implementation, and testing. J. Syst. Softw. 101, 136–146. http://dx.doi.org/10.1016/j.jss.2014.12.010.

Hsieh, H.-F., Shannon, S.E., 2005. Three approaches to qualitative content analysis. Qual. Health Res. 15 (9), 1277–1288. http://dx.doi.org/10.1177/1049732305276687.

IEEE, 2014. Software Engineering Competency Model (SWECOM). IEEE, http://www.dahlan.web.id/files/ebooks/SWECOM.pdf.

Kalliamvakou, E., Member, S., Bird, C., Zimmermann, T., Begel, A., Deline, R., German, D.M., 2019. What makes a great manager of software engineers? IEEE Trans. Softw. Eng. 45 (1), 87–106.

Kitchenham, B., Charters, S., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Kobata, K., Uesugi, T., Adachi, H., Aoyama, M., 2013. Software engineering education program for software professionals of high competency at. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), Vol. 2. pp. 117–122. http://dx.doi.org/10.1109/APSEC.2013.125.

Kropp, M., Meier, A., Perellano, G., 2016. Experience report of teaching agile collaboration and values agile software development in large student teams. In: 2016 IEEE 29th International Conference on Software Engineering Education and Training. CSEET, pp. 76–80. http://dx.doi.org/10.1109/CSEET.2016.30.

Le Deist, F.D., Winterton, J., 2005. What is competence? Hum. Resour. Dev. Int. 8 (1), 27–46. http://dx.doi.org/10.1080/1367886042000338227.

Lenberg, P., Feldt, R., Wallgren, L.G., 2015. Behavioral software engineering: A definition and systematic literature review. J. Syst. Softw. 107, 15–37. http://dx.doi.org/10.1016/j.jss.2015.04.084.

León-sigg, M. De, Pérez-valenzuela, B.J., Vázquez-reyes, S., Cisneros, J.L.V., 2018. Adaptation of the initial software development method for a single developer. In: 6th International Conference in Software Engineering Research and Innovation. http://dx.doi.org/10.1109/CONISOFT.2018.00013.

Li, Z., Avgeriou, P., Liang, P., 2015. A systematic mapping study on technical debt and its management. J. Syst. Softw. 101, 193–220. http://dx.doi.org/10.1016/j.jss.2014.12.027.

Licorish, S.A., Macdonell, S.G., 2013. Differences in jazz project leaders ' competencies and behaviors: A preliminary empirical investigation. In: 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering. CHASE, pp. 1–8. http://dx.doi.org/10.1109/CHASE.2013.6614725.

Linck, B., Ohrndorf, L., Schubert, S., Stechert, P., Magenheim, J., Nelles, W., Neugebauer, J., Schaper, N., 2013. Competence model for informatics modelling and system comprehension. In: 2013 IEEE Global Engineering Education Conference. EDUCON, pp. 85–93. http://dx.doi.org/10.1109/EduCon.2013.6530090.

Mead, N.R., Shoemaker, D., 2013. The software assurance competency model: A roadmap to enhance individual professional capability. In: Software Engineering Education Conference, Proceedings. pp. 119–128. http://dx.doi.org/10.1109/CSEET.2013.6595243.

Molléri, J.S., Petersen, K., Mendes, E., 2018. CERSE - Catalog for empirical research in software engineering: A systematic mapping study. In: Information and Software Technology. pp. 1–33. http://dx.doi.org/10.1016/j.infsof.2018.08.008.

Moreno, A.M., Sanchez-segura, M., Medina-dominguez, F., Carvajal, L., 2012. Balancing software engineering education and industrial needs. J. Syst. Softw. 85 (7), 1607–1620. http://dx.doi.org/10.1016/j.jss.2012.01.060.

Moustroufas, E., Stamelos, I., Angelis, L., 2015. Competency profiling for software engineers: Literature review and a new model. In: Proceedings of the 19th Panhellenic Conference on Informatics. pp. 235–240. http://dx.doi.org/10.1145/2801948.2801960.

Orsoni, A., Colaco, B., 2013. A competency framework for software development organizations. In: 2013 UKSim 15th International Conference on Computer Modelling and Simulation. pp. 507–511. http://dx.doi.org/10.1109/UKSim.2013.101.

Palvia, P., Leary, D., Mao, E., Midha, V., Pinjani, P., Salam, A.F., 2004. Research methodologies in MIS: An update. In: Communications of the Association for Information Systems 14(November). http://dx.doi.org/10.17705/1cais.01424.

Papoutsoglou, M., Kapitsaki, G.M., Mittas, N., 2018. Linking personality traits and interpersonal skills to gamification awards. In: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications. pp. 214–221. http://dx.doi.org/10.1109/SEAA.2018.00042.

Pérez, J., Vizcarro, C., García, J., Bermúdez, A., Cobos, R., 2017. Development of procedures to assess problem-solving competence in computing engineering. IEEE Trans. Educ. 60 (1), 22–28. http://dx.doi.org/10.1109/TE.2016.2582736.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12Th International Conference on Evaluation and Assessment in Software Engineering, Vol. 17. p. 10. http://dx.doi.org/10.1142/S0218194007003112.

Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18. http://dx.doi.org/10.1016/j.infsof.2015.03.007.

Radermacher, A., Walia, G., Knudson, D., 2014. Investigating the skill gap between graduating students and industry expectations. In: Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014. pp. 291–300. http://dx.doi.org/10.1145/2591062.2591159.

Rainsbury, E., Hodges, D., Burchell, N., Lay, M., 2002. Ranking workplace competencies: Student and graduate perceptions. pp. 8–18.

Rivera-Ibarra, J.G., Rodríguez-Jacobo, J., Serrano-Vargas, M.A., 2010. Competency framework for software engineers. In: 2010 23rd IEEE Conference on Software Engineering Education and Training. pp. 33–40. http://dx.doi.org/10.1109/CSEET.2010.21.

Robal, T., Ojastu, D., Kalja, A., Jaakkola, H., 2015. Managing software engineering competences with domain ontology for customer and team profiling and training. In: 2015 Portland International Conference on Management of Engineering and Technology. PICMET, pp. 1369–1376. http://dx.doi.org/10.1109/PICMET.2015.7273171.

Saiedian, H., Dale, R., 2000. Requirements engineering: Making the connection between the software developer and customer. Inf. Softw. Technol. 42, 419–428.

Saldaña-Ramos, J., Sanz-Esteban, A., García-Guzmán, J., Amescua, A., 2012. Design of a competence model for testing teams. IET Softw. 6 (5), 405–415. http://dx.doi.org/10.1049/iet-sen.2011.0182.

Salviano, C.F., Alves, A.M., Stefanuto, G.N., Maintinguer, S.T., Mattos, C.V., Reuss, G., Tecnologia, C. De, Archer, R., I, R.D.P., 2012. Developing a process assessment model for technological and business competencies on software development. In: 2012 Eighth International Conference on the Quality of Information and Communications Technology, Vol. 15504. pp. 125–130. http://dx.doi.org/10.1109/QUATIC.2012.27.

Schulte, C., Magenheim, J., Muller, K., Budde, L., 2017. The design and exploration cycle as research and development framework in computing education. In: 2017 IEEE Global Engineering Education Conference. EDUCON, pp. 867–876. http://dx.doi.org/10.1109/EDUCON.2017.7942950.

Sedelmaier, Y., Landes, D., 2013. A research agenda for identifying and developing required competencies in software engineering. International Journal of Engineering Pedagogy 3 (2), 0–4.

Sedelmaier, Y., Landes, D., 2014a. A multi-perspective framework for evaluating software engineering education by assessing students ' competencies SECAT – A software engineering competency assessment tool. In: 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. pp. 1–8. http://dx.doi.org/10.1109/FIE.2014.7044331.

Sedelmaier, Y., Landes, D., 2014b. Software engineering body of skills (SWEBOS). In: IEEE Global Engineering Education Conference. EDUCON, pp. 395–401. http://dx.doi.org/10.1109/EDUCON.2014.6826125.

Seppanen, V., 2000. A relationship-based view to software engineering competence. In: International Conference on Product Focused Software Process Improvement. pp. 376–390. http://dx.doi.org/10.1007/b72823.

Shaw, M., 2003. Writing good software engineering research papers. In: 25th International Conference on Software Engineering, 2003. Proceedings, Vol. 6. pp. 726–736. http://dx.doi.org/10.1109/ICSE.2003.1201262.

Stepanenko, V., Kashevnik, A., 2017. Competence management systems in organisations: A literature review. In: Proceeding of the 20th Conference of Fruct Association, pp. 427–433.

Studt, R., Winterfeldt, G., Mottok, J., 2015. Measuring software engineering competencies. In: IEEE Global Engineering Education Conference. EDUCON, pp. 908–914. http://dx.doi.org/10.1109/EDUCON.2015.7096081.

Tahvanainen, S., Luoma, E., 2018. Examining the competencies of the chief digital officer. In: Twenty-Fourth Americas Conference on Information Systems 2009, pp. 1–10.

Theisen, C., Dunaiski, M., Williams, L., Visser, W., 2017. Writing good software engineering research papers: Revisited. In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion. ICSE-C, p. 402. http://dx.doi.org/10.1109/ICSE-C.2017.51.

Thurner, V., Schlierkamp, K., Bottcher, A., Zehetmeier, D., 2016. Integrated development of technical and base competencies: Fostering reflection skills in software engineers to be. In: IEEE Global Engineering Education Conference. EDUCON, pp. 340–348. http://dx.doi.org/10.1109/EDUCON.2016.7474576.

Trivellas, P., Reklitis, P., 2014. Leadership competencies profiles and managerial effectiveness in Greece. In: Procedia Economics and Finance 9(Ebeec 2013). pp. 380–390. http://dx.doi.org/10.1016/S2212-5671(14)00039-2.

Tuffley, D., 2012. Optimising virtual team leadership in global software development. IET Softw. 6 (2011), 176–184. http://dx.doi.org/10.1049/iet-sen.2011.0044.

Turley, T., Bieman, M., 1995. Competencies nonexceptional of exceptional and software engineers. J. Syst. Softw. 28 (28), 19–38, (1995).

Urs, D.S., 2013. Soft skills for the engineering students. Synergy 9 (2), 137–142, https://pdfs.semanticscholar.org/fbf4/0a9446331b41d3f11ef3b7035fe33e2b452f.pdf.

Vaismoradi, M., Turunen, H., Bondas, T., 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. Nurs. Health Sci. 15, 398–405. http://dx.doi.org/10.1111/nhs.12048.

Vishnubhotla, S.D., Mendes, E., Lundberg, L., 2018. An insight into the capabilities of professionals and teams in agile software development A systematic literature review. In: ICSCA 2018, pp. 10–19.

Wernerfelt, B., 1984. A resource-based view of the firm. Strateg. Manag. J. 5 (2), 171–180. http://dx.doi.org/10.1002/smj.4250050207.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in software engineering. In: Experimentation in Software Engineering, Vol. 9783642290. http://dx.doi.org/10.1007/978-3-642-29044-2.

Nana Assyne is a doctoral student in computer science at the University of Jyväskylä, Faculty of Information Technology. He received his M.Sc. in Information Technology from the Lappeenranta University of Technology – Finland (2012) and received his bachelor's degree in Business Information Technology from Haaga-Helia University of Applied Sciences – Finland (2008). He also holds a professional certificate in pedagogical studies in vocational teacher education from Haaga-Helia University of Applied Sciences – Finland (2019). Prior to his Ph.D. studies, he had over 5 years of teaching experience in software engineering courses at the university level. Current research areas include software engineering competencies and software startup engineering.

Hadi Ghanbari is a postdoctoral researcher at the Department of Information and Service Management, Aalto University School of Business, Finland. Simultaneously, he is a visiting research fellow at FinEst Twins Smart City Center of Excellence, TalTech, Estonia. He mostly conducts empirical research on sociotechnical aspects of information systems development and digital innovation, with a focus on Technical Debt and information security. He holds a Ph.D. in Computer Science from the University of Jyväskylä, Finland (2017), a M.Sc. in Information Systems from the University of Oulu, Finland (2012), and a B.Eng. in Software Engineering from Azad University, Iran (2005). During his academic career, he has been involved in several industry-driven R&D projects both as a project manager and a researcher. Before his academic career, he has worked in several positions in the IT industry for more than eight years.

Mirja Pulkkinen works at the University of Jyväskylä, Faculty of IT, since 2011 as teacher of Information Systems. With an MA degree earned earlier, she joined the Faculty 2001 to work in collaborative industry research projects and completed a M.Sc. degree in Information Systems 2005. In research, her focus has been enterprise architecture and organization IT management, and she defended a dissertation in 2008 on this area. Her further research concerns software engineering and SPI, as well as IT in Education, which follows her interest and experience in an earlier teaching career involving e-learning research and development. As a research project manager, she has been engaged in projects funded by the National Funding Agency for Science and Technology (later Business Finland), the EU Commission, and other sponsors.