WAGENINGEN
UNIVERSITY & RESEARCH

Aligning software engineering education with industrial needs: A meta-analysis

Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M.

Please cite this publication as follows:

Aligning software engineering education with industrial needs: a meta-analysis

Vahid Garousi
Information Technology Group
Wageningen University, Netherlands
vahid.garousi@wur.nl

Görkem Giray
Independent Researcher, Turkey
gorkemgiray@gmail.com

Eray Tüzün
Department of Computer Engineering
Bilkent University, Turkey
eraytuzun@cs.bilkent.edu.tr

Cagatay Catal
Information Technology Group
Wageningen University, Netherlands
cagatay.catal@wur.nl

Michael Felderer
University of Innsbruck, Austria &
Blekinge Institute of Technology, Sweden
michael.felderer@uibk.ac.at

Abstract:

Context: According to various reports, many software engineering (SE) graduates often face difficulties when beginning their careers, which is mainly due to misalignment of the skills learned in university education with what is needed in the software industry.

Objective: Our objective is to perform a meta-analysis to aggregate the results of the studies published in this area to provide a consolidated view on how to align SE education with industry needs, to identify the most important skills and also existing knowledge gaps.

Method: To synthesize the body of knowledge, we performed a systematic literature review (SLR), in which we systematically selected a pool of 35 studies and then conducted a meta-analysis using data extracted from those studies.

Results: Via a meta-analysis and using data from 13 countries and over 4,000 data points, highlights of the SLR include: (1) software requirements, design, and testing are the most important skills; and (2) the greatest knowledge gaps are in configuration management, SE models and methods, SE process, design (and architecture), as well as in testing.

Conclusion: This paper provides implications for both educators and hiring managers by listing the most important SE skills and the knowledge gaps in the industry.

Keywords:

Software engineering education; industry needs; important skills; knowledge gap; systematic literature review (SLR); meta-analysis

TABLE OF CONTENTS

1 INTRODUCTION

The quality of the software engineering (SE) workforce is a direct result of the quality of SE education, and how SE students are trained. However, there is a widely-noted "gap" between software industry's skill needs and the education university students receive [1, 2]. While most university programs focus mainly on computer science (CS) core topics, SE topics get less attention [1]. As early as in 1989, a paper by Ford and Gibbs [3] succinctly stated a problem that still partially exists today: "*Somewhat oversimplified, industry needs software engineers, but universities are supplying computer scientists. Thus, it's time to promote widespread development of software engineering degree programs*".

It has been reported that many recent SE graduates often face difficulties when beginning their professional careers [1, 2]. Some in the community believe that: "*The software engineering shortage is not a lack of individuals calling themselves "engineers", the shortage is one of quality – a lack of well-studied, experienced engineers with a formal and deep understanding of software engineering*" [4]. It is widely accepted that better education of SE students will better prepare them for their SE careers after graduation and will increase their employability [5]. An effective approach to close the gap between academia and industry in SE education and to best train SE students is for both parties to work together on educational needs and goals [6]. If planned properly, such a collaboration can meet industry's education and training goals.

To address the above need, many studies (35 papers according to our literature search) have been conducted to "align" SE education with industry's skill needs. Each of those studies has focused on a regional and limited-scale dataset, usually gathered via opinion surveys. Thus, it is important to synthesize and aggregate the results of all those studies to provide a single "consolidated" view on how to align SE education with industry needs, and to find out the most important skills in the industry and the knowledge gaps, i.e., the SE topics that should get more education and training. As per the principles of meta-analysis [7], combining and synthesizing data and evidence from all previous studies will provide a better and more comprehensive view on the topic.

To further motivate the need for such a meta-synthesis, the authors themselves are all active SE educators and have been teaching various SE courses for more than 15 years each. The authors have been involved in SE education research, e.g., [8-13]. Furthermore, they have had active industry experience or have worked in close collaborations with practitioners in many joint industry-academia collaboration projects, e.g., [14, 15] and have observed how engineers utilize their skills to conduct SE tasks and how lack of SE expertise could challenge an engineer doing her/his tasks. According to feedback from our industry partners who have hired our students, feedback from our recently-graduated students and needs of our university departments and SE programs, we have seen the need to synthesize the findings of studies in this area to best align SE education activities with industrial needs.

To conduct the analysis reported in this paper, we used the established guidelines and processes for performing systematic literature review (SLR) studies [16] and meta-analysis [7] in SE. Our systematic paper search and selection phase resulted in a set of 35 papers on this subject published between 1995 and 2018.

Our goal in this paper is to shed light on the importance and knowledge gaps of different SE topics and to answer the question of how to best train software engineers of tomorrow. By summarizing what we as a community know in that area, our article aims to benefit the readers (both educators and hiring managers) by providing both a "big picture" on the state of the community w.r.t. aligning SE education with industrial needs, and an "index" to the body of knowledge and evidence in that area. Furthermore, "*because it is often impractical for readers to track down and review all of the primary studies [in this area], review articles [such as this work] are an important source of summarized evidence on a particular topic*" [17].

The remainder of this article is structured as follows. Section 2 provides a review of related work. Section 3 describes the research method and the review planning. Section 4 presents the search phase and the selection of the pool of sources to be reviewed. Section 5 discusses the development of the systematic map and data-extraction plan. Section 6 presents the results of the literature review. Section 7 summarizes the findings and discusses the lessons learned. Finally, in Section 8, we draw conclusions and suggest areas for further research.

2 A REVIEW OF RELATED WORKS

SE education and training are active areas among the "education research" and practitioners communities. The flagship conference in this area is the IEEE Conference on Software Engineering Education and Training (CSEE&T), which was held for the 31th year in 2019. There are also various Computer Science (CS) education conferences, such as the ACM technical symposium on CS education (SIGCSE) which was held for its 50th year in 2019.

There is a very large number of papers in SE education, a large portion of which have been published in the above two venues (CSEE&T and SIGCSE). Since our goal in this work is to conduct a meta-analysis, the related works in our context are other *survey/review* papers (secondary studies) in this area. We searched for secondary studies in SE education and we were able to find many such studies, e.g. [1, 18-24].

Out of those many secondary studies, we found only four studies [1, 20, 22, 23] having RQs about "aligning" SE education with industrial needs, and we list those studies in Table 1. For example, one of the RQs in [1] was to determine: *Is there empirical evidence of knowledge deficiencies in graduating computer science students?*

For each review study in Table 1, we include its publication year, the number of papers reviewed by the study, and the related research questions (RQs) studied in each paper. As one can see in Table 1, the earliest survey paper in this area was published in 2012.

We have added to Table 1 a column in which we objectively discuss the limitations of each of the related studies and how our work addresses that limitation. In addition to those limitations, and also in addition to reporting knowledge gaps (our RQ 4), this meta-study provides extra novelty / insights w.r.t. the importance of SE skills (our RQ 3 in Section 6.4). We should highlight that, as one RQ of our study (RQ 8), we will compare our results to findings of those review papers when the data are comparable (in Section 6.8).

Table 1: A list of other survey (review) papers on aligning SE education with industrial needs

Paper title	Year	Reference	Type of secondary study	Num. of papers reviewed	RQs / objectives related to our scope (aligning SE education with industrial needs)	Limitations of the study / How our work addresses that limitation
Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectations of industry	2012	[1]	Systematic literature review (SLR)	28	• RQ1: <u>Is there empirical evidence of knowledge deficiencies in graduating computer science students?</u> • RQ2: How do the knowledge deficiencies	The body of knowledge used in the study was ad-hoc (not an established one), while we are using the Software Engineering Body of Knowledge (SWEBOK) v3 [25]. The study marked whether each primary study mentioned the knowledge deficiency a SE topic

					that have been identified by academia and industry differ? • RQ3: Can a classification system for the identified knowledge deficiencies be created?	as a Boolean (yes/no), while our study uses the relative ranking data from the primary studies (see Section 6.4). Such ranking data would provide more precise aggregate data at the end of the meta-analysis.
Gaps between industry expectations and the abilities of graduates	2013	[22]	SLR	30	• RQ1: Is there empirical evidence of knowledge deficiencies in graduating computer science students? • RQ2: What are the most common knowledge deficiencies in students? • RQ3: Are there trends or changes in knowledge deficiencies in students over time?	The same limitations as discussed for [1] above
A systematic mapping study on practical approaches to teaching software engineering	2014	[20]	Systematic mapping	173	• RQ1: What are the main approaches used to address the practical experiences in software engineering education? • RQ2: Is there an emerging tendency to address this challenge [practical experiences]? • RQ3: Which software process models are used to support the practical experiences in software engineering courses? • RQ4: Have the universities changed the way of conducting these experiences over the years?	The same limitations as discussed for [1] above
Preparing ICT graduates for real-world challenges: results of a meta-analysis	2017	[23]	Meta-synthesis	155	• RQ1: Does the paper stress the need for changes in curriculum design and delivery? • RQ2: Does the paper stress the mismatch between learning outcomes of ICT graduates and skills requirements from the labor market? • RQ3: Does the paper stress the need for different/new teaching and assessment methods?	The RQ 2 was addressed very briefly and only with general non-quantitative results. Our SLR synthesizes knowledge gap data quantitatively.
Closing the gap between software engineering education and industrial needs	2019	[24]	Meta-synthesis	33	The short version of the work reported in this paper, published in IEEE Software, which reported only three of the RQs, and in in less detail and more from a practitioner's perspective compared to this paper	

It should be mentioned that a short version of the current work has recently been accepted for publication in IEEE Software [24]. The main difference between the paper at hand and the short version published in IEEE Software is that the this full paper targets researchers and SE educators, whereas the IEEE Software article [24] targeted mainly practitioners. Therefore, in the IEEE Software article we reported only the following three of the overall eight RQs, and also covered them in less detail and more from a practitioner's perspective compared to this full paper:

• What skills are the most important in the software industry? (RQ 3)
• Is there evidence of knowledge deficiencies in graduating SE students? (RQ 4)
• To what extent are soft skills important, in addition to hard (technical) skills? (RQ 5)

3 RESEARCH METHOD AND SLR PLANNING

In this section, we present an overview of our research method (Section 3.1) and then the goal and research questions of our study (Section 3.2). The authors should note that since SLRs in SE has a standard and established process [16], the process and text of the current Section 3 and Sections 4 and 5 are quite similar to the other SLRs published by other researchers and by the authors in the past, e.g., [26, 27].

3.1 OVERVIEW

We developed the SLR process as illustrated in Figure 1. The process was developed based on our past experience in systematic mapping (SM) and SLR studies, e.g., [26, 27], and the well-known guidelines for conducting literature reviews and meta-analysis in SE (e.g., [7, 16, 28-30]). As shown, part of the work is a SM in which we categorized the primary studies w.r.t. a set of defined attributes. Another component of our work is the meta-analysis of quantitative data about the importance of SE skills, and knowledge gaps, as extracted from the primary studies.

The next subsection gives an overview of the design of SLR along with its goal and RQs. The subsequent phases are presented in the Sections 4 to 6.
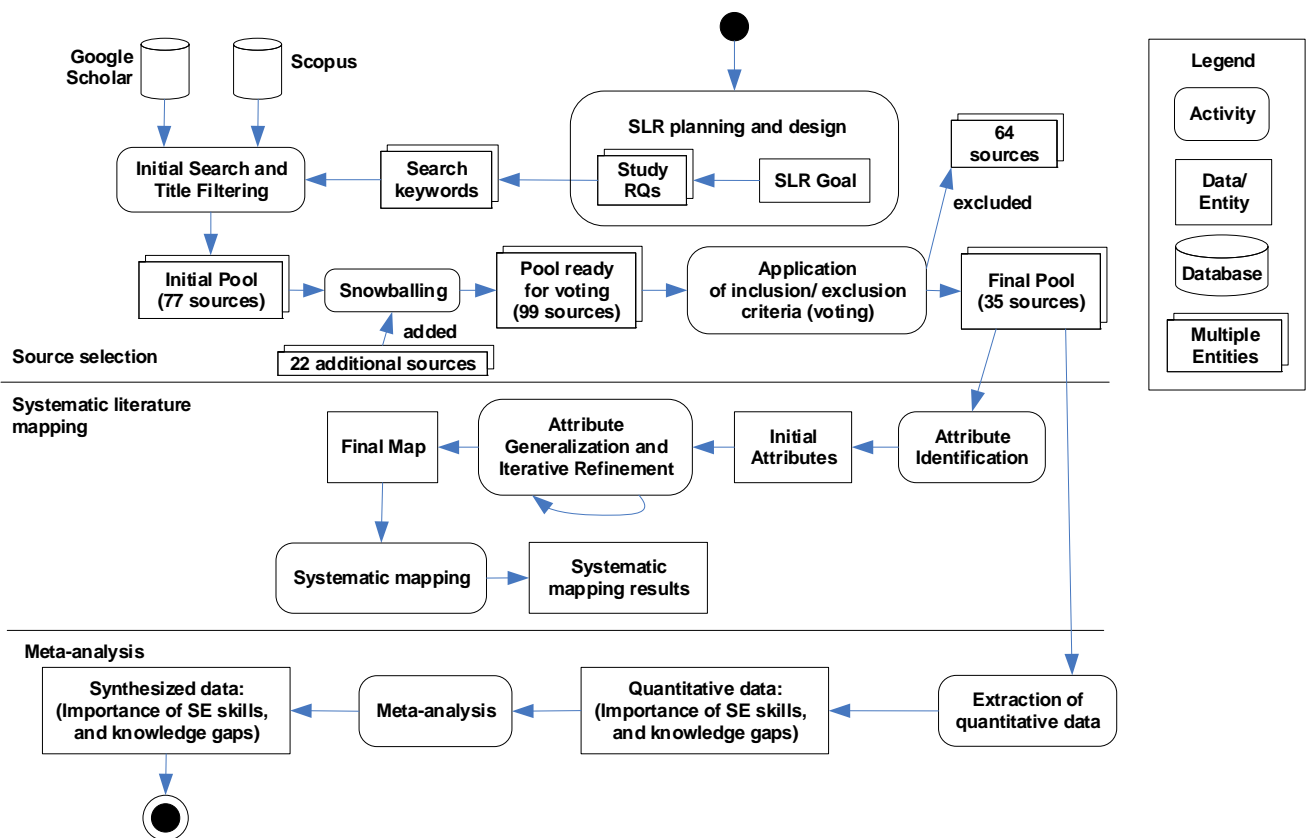
Figure 1-An overview of the applied SLR process (represented as a UML activity diagram), similar to the process in our previous SLRs [31]

3.2 GOAL AND RESEARCH QUESTIONS

The goal of this study is to systematically classify, review, and synthesize the body of knowledge and evidence in the scope of aligning SE education with industrial needs, from the point of view of educators and hiring managers. We raised the following research questions (RQs) to achieve our goal:

- **RQ 1- Types of data collection methods and metrics**: What types of data collection methods have been used in the studies? We observed that while some studies use opinion surveys to gather data, others use interviews or mine job advertisements to extract the data to conduct their analysis. Also, what metrics have been used? Some related metrics are: how much a given subject has been learned by students in university programs, and how much engineers currently know about a given subject?

- **RQ 2- Curriculum models (bodies of knowledge)**: What curriculum models (bodies of knowledge) have been used to design the studies? Some examples are the Software Engineering Body of Knowledge (SWEBOK) [25], the Software Engineering Education Knowledge (SEEK) [32], and the ACM SE 2004 curriculum guideline [33].
- **RQ 3- Importance of SE skills**: Which SE skills (subjects) are the most important and which SE subjects are the least important for a typical software engineer in practice?
- **RQ 4- Knowledge gaps (deficiencies)**: What is the evidence for knowledge gaps (deficiencies) in different SE subjects? And what are the topics with the highest knowledge deficiencies?
- **RQ 5**- **Importance of soft skills**: What is the evidence on the importance of soft skills? And to what extent are soft skills important, in addition to technical ("hard") skills?
- **RQ 6- Dataset (data sample) characteristics**: What are the characteristics of the data sample? For example, what is the number of data points (e.g., participants, job advertisements) in the survey conducted by a given study? Which region (countries) are covered by the surveys?
- **RQ 7- Synthesis of educational recommendations/ findings**: What educational recommendations are provided in each study? We wanted to synthesize such recommendations for the benefit of educators and hiring managers.
- **RQ 8- Comparison of our results to previous review papers**: How do the findings of our meta-analysis (importance of SE skills, and knowledge gaps) compare to the findings of previous review papers [1, 22, 23] (which had similar objectives), as discussed in Section 2?

We should note that a shorter version of this work, covering only RQs 3-5, has been prepared as a short paper [24]. Thus, the current paper is a "major" extension of that short paper.

4 SEARCHING FOR AND SELECTION OF PRIMARY STUDIES

As Figure 1 illustrates, we searched for and selected primary studies in the first phase, which consists of three steps, as discussed next. Section 4.1 presents source selection and search keywords. Section 4.2 summarizes how we applied inclusion and exclusion criteria, and Section 4.3 gives an overview of the final pool of papers and the online repository.

4.1 SELECTING THE SOURCE ENGINES AND SEARCH KEYWORDS

We performed the searches in two online academic search engines (paper databases): Google Scholar and Scopus (www.scopus.com), which are widely used in SLR studies, e.g., [34, 35]. The reason that we used Scopus along with Google Scholar was that several sources have mentioned that: "*it [Google Scholar] should not be used alone for systematic review searches*" [36] as it may not be sufficient to find all relevant papers. Several SLRs in SE also use other digital libraries such as ACM Digital Library, IEEE Xplore, Springer, and ScienceDirect. There are empirical studies in information and library sciences, such as [37], which have shown that Google Scholar indexes and sufficiently covers other well-known digital libraries such as ACM Digital Library, IEEE Xplore, Springer, and ScienceDirect. To validate our search strategy and to reduce threats to validity we conducted our own validation and re-executed the searches with our keywords (discussed below) in ACM and IEEE Xplore. We then checked the first 5 pages of each result set and did not find any additional papers not yet included. Thus, it made sense to use the "subsuming" (larger) index which anyways would find items in each of the above individual digital libraries.

All authors conducted searches in Google Scholar and Scopus using the search string independently. When searching, the authors decided on which papers to include by applying inclusion and exclusion criteria. The search string was: *(educational needs OR knowledge needs OR desired skills OR essential competencies OR knowledge requirements OR skill requirements) AND (software engineers OR software developers)*.

The searches were conducted in winter 2018. Data extraction from the primary studies and their classifications were conducted during the winter and spring 2018. We wanted to include all the papers on this topic, thus we did not restrict the papers' year of publication (e.g., only those after 2000).

To ensure the inclusion of directly- or potentially-relevant papers, we applied title filtering to make our paper search and selection efforts efficient. To this end, we used the first inclusion/exclusion criterion (i.e., Does the source focus on aligning SE education with industrial needs?). Figure 2 illustrates a sample screenshot of some results we obtained from Google Scholar. As is seen from Figure 2, some results are directly- or potentially-relevant (highlighted by green boxes) and some are the clear out-of-scope papers (highlighted by red boxes). To best manage the efficiency of our efforts, we only added the related studies to the candidate pool. We had taken a similar approach in our provost SM and SLR studies [27, 38-42].

Another issue was the stopping condition when searching using Google Scholar. As Figure 2 shows, Google Scholar provided a very large number of results using the search string (more than 200,000 records). It is impossible to go over all of the results. We utilized the relevance ranking of Google Scholar (Google's PageRank algorithm) to restrict the number of

results. We checked the results in the first n pages and stopped checking when we could not find any relevant paper in the nth page. Several other review studies, guideline and experience papers [43-47] have reported the use of a similar heuristics. As Figure 1 shows, our initial pool consisted of 77 papers in consequence of our initial search and title filtering.

Also, since both Scopus and Google Scholar were used, there were chances of duplications in the pool. We only added to the pool a candidate paper if it was not already in the candidate pool.
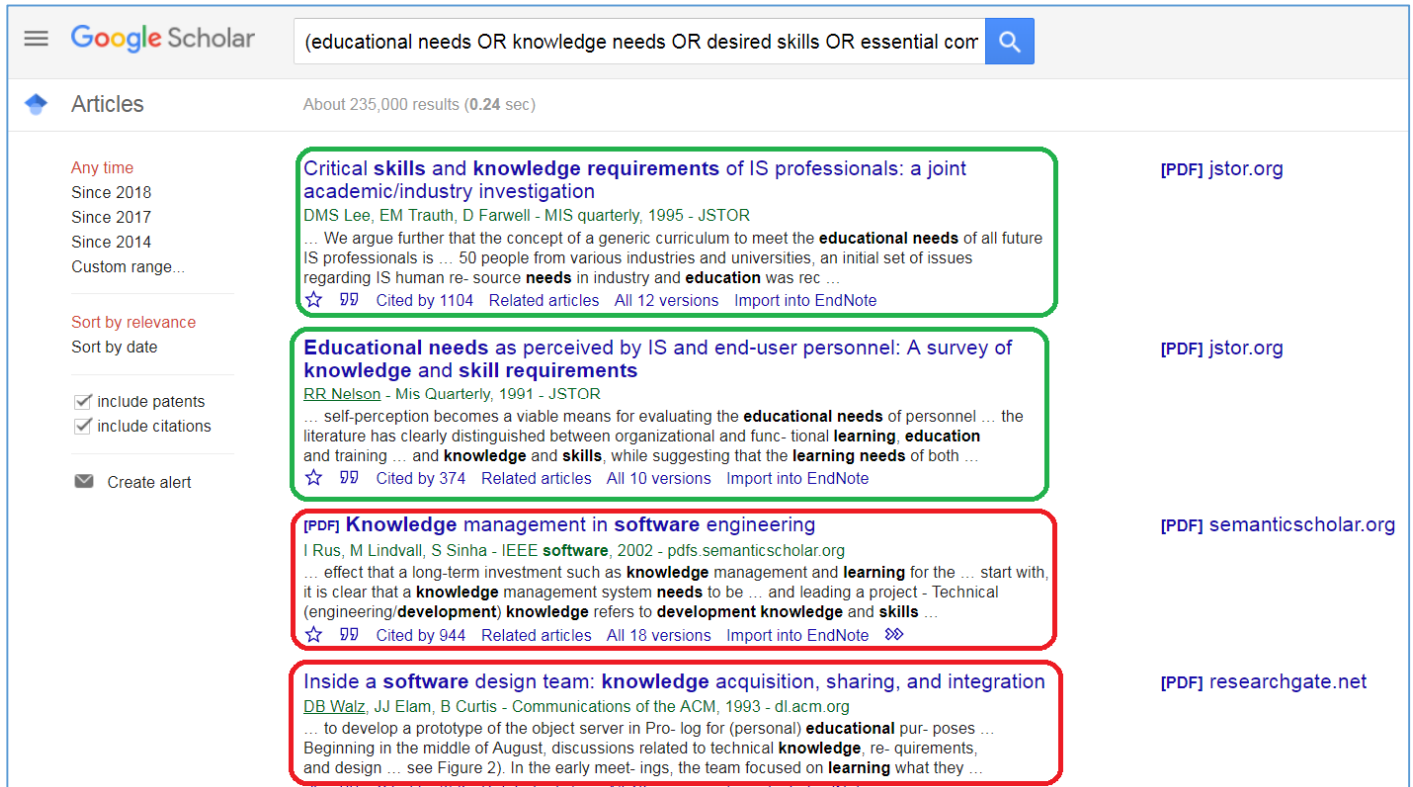

Figure 2- A screenshot from the search process using Google Scholar

As recommended by systematic review guidelines, we also conducted forward and backward *snowballing* [29] on the papers already in the pool, to include all the relevant sources as much as possible. Backward snowballing involves the application of inclusion/exclusion criteria to the reference list of a paper that is already in the pool. In forward snowballing, inclusion/exclusion criteria are applied for the papers citing a paper that is already in the pool. Both forms of snowballing are often used in SLR/SM studies for identifying further related papers [29].

We found 22 additional papers using snowballing. For instance, paper [P4] was found by "forward" snowballing of [P33]. Papers [P31] and [P32] were found by forward snowballing of [P1]. Note that the citations in the form of [Pn] refer to the IDs of the primary studies (papers) reviewed in our study. They are available in the study's online spreadsheet (goo.gl/HU9Wz4).

The initial pool was composed of 99 "candidate" papers. The authors conducted a systematic voting (as discussed next), in which a set of defined inclusion/exclusion criteria were applied to derive the final pool of the primary studies.

We should highlight that there is no perfect search strategy in SLRs, as we have learned in conducting and publishing several systematic literature reviews in software engineering, e.g., [26, 27]. We have made all our effort to minimize the chances of missing relevant papers, but of course there is no guarantee for that. This issue has been discussed elsewhere as well, e.g., in the medical sciences [48].

4.2 APPLICATION OF INCLUSION/EXCLUSION CRITERIA AND VOTING

We carefully defined the following inclusion criteria to ensure including all the relevant sources and not including the out-of-scope sources. The inclusion criteria were as follows:

- Does the paper focus on aligning SE education with industrial needs?
- Is the paper based on empirical data and not just the opinion of the authors?

- Is the source in English and can its full-text be accessed on the Internet?

The answer to each question could be binary: either Yes (value=1) or No (value=0). Our voting approach was as follows. One of the researchers voted on all the candidate papers w.r.t. all the above three criteria. The other researchers then peer-reviewed all those votes. We used a consensus approach where disagreements were discussed until consensus was reached among all authors.

We included only the sources which received 1's for all the above three criteria and excluded the rest. Application of the above criteria led to exclusion of 64 sources, details for which can be found in the study's online spreadsheet. For example, we excluded [49, 50] since they were not based on empirical data but rather based on only the opinions of the authors. Many papers had a remote relation to "SE" education and had not considered a reasonable set of SE topics, but rather had focused on IS (information systems) or IT education, which we excluded, e.g., [51, 52]. The entire list of excluded papers with their exclusion reasons are available in the "Excluded" tab the online spreadsheet (goo.gl/HU9Wz4).

4.3 FINAL POOL OF THE PRIMARY STUDIES

The list of the final pool of 35 papers can be found in an online spreadsheet at goo.gl/HU9Wz4. Once we finalized the pool of papers, we looked at the growth of this field by the number of published papers per year. We depict in Figure 3 the annual number of papers (by their publication years). Note that, as discussed in Section 4.1, since we searched for the papers during winter 2018, the number of papers for 2018 is partial (only one paper).

We see in Figure 3 that researchers have been active in the area since the mid 1990's and the annual number of papers had reached its maximum in the year 2015 (5 papers). Figure 3 shows that, although SE has been taught in many universities and colleges around the world for several decades now, SE educators still struggle to properly align SE education with industrial needs, as visible by the number of recent studies.
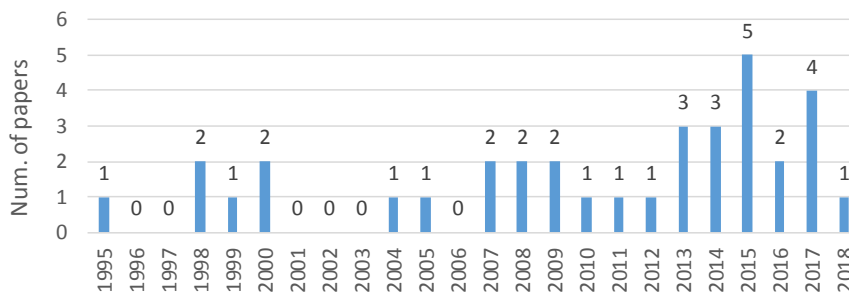

Figure 3-Growth of the field over the years

5 DEVELOPMENT OF THE SYSTEMATIC MAP AND META-ANALYSIS PLAN

As discussed in our SLR process (Figure 1), one of the two activities in our SLR is a SM in which we wanted to categorize the primary studies w.r.t. a set of defined attributes (RQs). To achieve this, we developed a systematic map and then extracted data from the papers to classify them using the map. In this section, we discuss the development of the systematic map (Section 5.1), and then the process for data extraction and meta-analysis [7] (Section 5.2).

5.1 DEVELOPMENT OF THE SYSTEMATIC MAP

The first step in developing our systematic map was the analysis of the studies in the pool and the identification of the initial list of attributes. We derived the final map by generalizing and refining the initial attributes, as shown in Figure 1.

We recorded the studies, which we identified as relevant, in our online spreadsheet to facilitate further analysis. Next, we categorized the studies to answer the RQs and build a current picture of the research area. We synthesized the findings of the primary studies and built a systematic map iteratively.

Table 2 shows the final systematic map (classification scheme) that we developed after applying the process described above. In the table, column 1 shows the study's RQs, column 3 shows the corresponding attribute/aspect of each RQ, and column 4 shows the set of all possible values for the attribute. Column 4 indicates, for an attribute, whether multiple or single selections of values can be applied. For example, for RQ 1 (types of data collection method), the corresponding value in the last column is "M" (Multiple), indicating that one study can contribute more than one data collection method (e.g., opinion survey, interviews, data from mining job advertisements, etc.). In contrast, for RQ 3 (base models used), the corresponding value in the last column is "S" (Single), indicating that one paper has used only one base model.

The last column in Table 2 shows the chosen research approach to answer each RQ. To find the suitable research approaches, we studied the guidelines for selecting empirical methods for SE research [53], and also the guidelines for performing SLR [16]. Due to the type of each RQ, we found that four of the eight RQs (as shown in the table) should be addressed by conventional systematic mapping (SM), based on attributes listed in Table 2.

For two RQs (RQ 3 and RQ 4), since our goal was to find the (degree of) evidence for importance of and knowledge gaps in SE skills, we applied meta-analysis [7] to extract and synthesize quantitative data from the primary studies.

For RQ 5 (important soft skills) and RQ 7 (educational recommendations), the data that we planned to extract were qualitative phrases from each paper. Given the nature of the extracted data for those two RQs, it was natural to choose qualitative coding [54] as our method of choice [53]. We will further discuss our meta-analysis and qualitative coding approaches in Section 5.3.

To derive the categories for all attributes/aspects in the systematic map (Table 2), we had in mind an initial list of categories. We then used attribute generalization and iterative refinement. If a category appeared in at least two papers, we included it as a separate category in the corresponding set, otherwise, we added it in the "Other" categories.

Table 2: Systematic map developed and used in our study

RQ	Attribute/Aspect	Categories	(M)ultiple/ (S)ingle	Research approach
1	Types of data collection method	{Opinion survey, Interviews, Data from mining job advertisements, Other}	M	SM
	Types of metrics used	{How much learned in university, how much a software engineer knows about a topic currently, How important (needed, relevance) a topic is, Usage and usefulness of the topic in work, Knowledge gap (deficiency)= learned-importance, Other}	M	
2	Base models (bodies of knowledge) used	{SWEBOK v 1.0 (1999), SWEBOK v2. (2004), SWEBOK v3 (2014), SEEK 2003, ACM SE 2004, ACM SE 2014, ACM computer engineering curricula (2004), ACM computer science curricula (CCCS) (2013), ACM IT curriculum (2017), ACM IS curriculum (2010), SE topics in the Lethbridge paper [P33], Synthesized from literature, Synthesized from initial interview data, Existing university courses, Other}	S	SM
3	Importance of SE skills	Quantitative importance data for SE skills categorized using 15 "knowledge areas" (KAs) of SWEBOK [25]: Requirements, design (and architecture), development (programming), testing, maintenance, configuration management, project management, SE process, SE models and methods, quality, SE professional practice, SE economics, computing foundations, engineering foundations, and mathematical foundations	M	Meta-analysis
4	Knowledge gaps (deficiencies)	Quantitative knowledge gaps data for SE skills categorized using 15 KAs of SWEBOK	M	Meta-analysis
5	Soft skills	Soft skill(s) deemed important, as mentioned in each paper: Qualitative data (string)	M	Qualitative coding
6	Data sample characteristics	• Number of data points (participants, job advertisements): integer • Region (or country) the survey was conducted in: Name	M	SM
7	Educational recommendations	Educational recommendations as mentioned in each paper: Qualitative data (string)	M	Qualitative coding
8	No data from the primary studies are needed for this RQ, since the RQ's goal is comparing our results to previous review papers.			

5.2 PROCESS FOR DATA EXTRACTION

After developing the systematic map, each of the authors extracted data from the subset of the papers assigned to him. To ensure post-verification and quality of our data extraction activity, the authors recorded in their online data spreadsheet comments on each extracted data item pointing to the exact phrases/data in each primary study (a form of "traceability"). This made it easy and systematic to determine (and justify) why each classification was performed.

We also conducted systematic peer reviewing after the authors finished their data extraction. Each author peer-reviewed the data extraction of another author in the team. In the case of disagreements, further discussions took place to reach a consensus. Such discussions ensured the quality and validity of our results.

5.3 PROCESS FOR DATA SYNTHESIS AND META-ANALYSIS

For two of the study's RQs (RQ 3 and RQ 4), we needed to conduct meta-analysis [7, 55, 56] to synthesize and consolidate the quantitative data about the importance of SE skills, and knowledge gaps, as extracted from the primary studies. Meta-analysis is "*a form of synthesis that combines the numerical results of primary studies*" [7]. We consulted and carefully followed the guidelines for meta-analysis in SE [7, 55, 56].

All papers had presented rankings for the most important skills and knowledge gaps. To be able to cross-compare and synthesize data in a consolidated manner, we harmonized those ranking data in our meta-analysis as follows.

We normalized the ranking of topics in each paper to the range of [0, 1], for each SWEBOK KA. For example, for [P1], three of the 13 ranked topics were related to the "design" KA: general architecture (in rank 1), object-oriented design (rank 9), and user-interface design (rank 12). We calculated the average of (1, 9, 12), which equals 7.33, and divided it by the number of all SE topics in that paper (13), and the normalized rank metric was 0.56. We show in Figure 4 the details for this example. We show a screenshot from the actual table in the primary study [P1] from which we extracted the data.

Note that since we used rank data, the lower the value of this metric, the higher the importance of a given topic. Thus, by calculating "*1- normalized rank*", we aggregated the normalized importance (=1-0.52=0.48, for the above example paper). Once we had the importance metric value of each KA for each paper, we calculated its average among all papers.
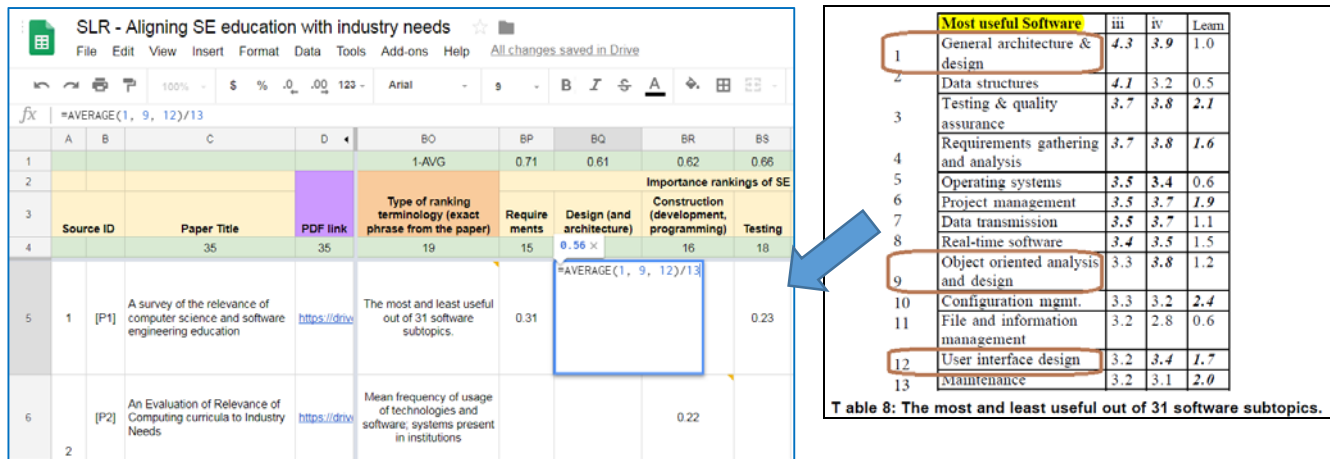
Figure 4- Right: An example screenshot from the primary study [P1] from which the quantitative data for the importance of SE skills were extracted. Left: A screenshot from the data sheet showing an example of how we stored the data for each paper and consolidated all the data.

Last but not the least, for RQ 5 (important soft skills) and RQ 7 (synthesis of educational recommendations), we used qualitative coding [54], which we have applied in several other recent SLR studies, e.g., [46, 47]. We essentially took at each qualitative piece of data about educational recommendations, as extracted from each paper, and then performed "open" and "axial coding" [54]. The creation of the new factors in the "coding" phase was an iterative and interactive process in which all five researchers participated. Figure 5 shows a screenshot for an example of how we conducted qualitative coding of educational recommendations. We will discuss more details and results of this process in Sections 6.5 and 6.7.

Figure 5- A screenshot showing an example of how qualitative coding of educational recommendations was conducted

6 RESULTS

In this section, we present the results of the study's research questions (RQs).

6.1 RQ 1: TYPES OF DATA COLLECTION METHODS AND METRICS USED

We show in Figure 6 the breakdowns of types of data collection methods and metrics used in the studies. 25 studies conducted opinion surveys to collect data from participants. Eight studies interviewed participants from both private and public sectors either face-to-face or via phone. [P16] formed "focus" groups to gather opinions via interviews.

Three studies [P16, P26, P35] mined job advertisements to explore skills desired by industry. For example, [P35] analyzed a set of 101 online job ads in the area of requirements engineering (RE) in the Netherlands to answer the question of: What the job market wants from requirements engineers?

Two studies used "other" data collection methods [P6, P29]. [P6] used a technique called "360-degree assessment" to provide a holistic view of employee skills in a multi-organizational study. Also known as 360-reviews or 360-degree employee-feedback, 360-assessments provides a holistic view of a set of employees through a process of gathering feedback from an employee's manager, peers and direct reports [57]. [P29] reported the findings of an observational study of eight recent college graduates in their first six months of a software development position at Microsoft Corporation.

As for all empirical studies, the data collection methods used in the papers suffered from certain threats including coverage, nonresponse, and measurement errors [58]. Many papers in the pool actually mentioned this explicitly. Coverage errors occur when the participants do not include some segments of the target population [58]. All of the studies gathered data from a limited region with only alumni of one or a few universities. Moreover, some of the studies targeted specific SE roles, such as systems analysts, business analysts and project managers [P9], and real-time and embedded software developers [P1]. Some of the studies reported low response rates, e.g., [P1, P3, P4, P8], which can cause bias in the results.

11

Measurement errors arise when participants answer inaccurately due to the wording of questions, order of questions, interviewer effect[1], or other external factors [58]. For instance, [P1] reported a possible difference between feeling and actual usefulness. Our meta-analysis study hopes at decreasing these threats by combining the results of all the studies.

In terms of the metrics used in the studies, as Figure 8 illustrates, most of the studies (24 papers) quantified the importance of SE skills. Eight studies measured the knowledge gaps (skills required by industry, but which SE graduates do not possess sufficiently). Four studies measured how much SE graduates learned in university, two studies measured how much they know about SE topics, and two studies evaluated the usefulness of skills in industry. In most cases, these metrics were assessed using Likert scales. While all of the studies aimed to contribute to aligning SE education with industry needs in general, these metrics have been used to address various research questions. For example, [P6, P9] compared the importance of/deficiency in soft and hard skills. [P7] identified the importance of skills according to different positions. [P11, P12] reported on importance perception differences between employers and graduates.
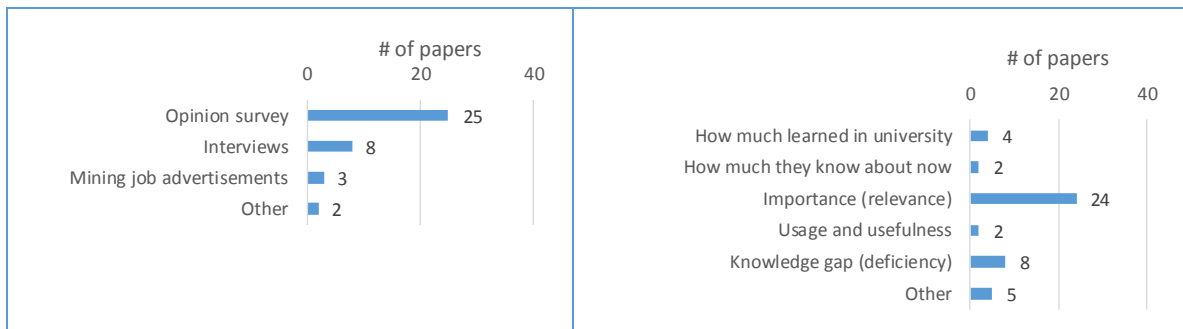
Figure 6- Types of data collection methods (left) and metrics used (right)

6.2 RQ 2: BASE MODELS (BODIES OF KNOWLEDGE) USED FOR CLASSIFICATION OF SE TOPICS

For classification of SE topics for the purpose of aligning them with industrial needs, studies used or proposed various classification models. The breakdown of those base models is shown in Figure 7.

There have been many efforts towards classifying SE topics, which are often known as "bodies of knowledge". The Software Engineering Body of Knowledge (SWEBOK) is one of those community-based attempts for constructing a taxonomy of SE topics. Since the SE discipline is continuously evolving due to emerging new technologies and advances in practice, SWEBOK is regularly updated with new versions: its version #1 was released in 1999 [59], version #2 in 2004 [60], and version #3 in 2014 [25].

Three studies [P1, P3, P33] used the topic taxonomy published in SWEBOK v1 and three [P4, P6, P14] using SWEBOK v2. None of the studies used SWEBOK v3, which is the most updated body of knowledge for SE. Our study fills this gap by re-classifying and combining all the skills reported in 35 studies based on SWEBOK v3.

In addition to SWEBOK, some other efforts have been made under the umbrella of ACM/IEEE and have proposed guidelines for curriculum design, namely, the ACM "SE 2004" guideline [33] and "SE 2014" [61], ACM Computer Engineering Curricula 2004 [62] and 2016 [63], ACM Computer Science Curricula 2013 [64], ACM IT curriculum 2017 [65], and ACM IS curriculum 2010 [66]. 10 studies, in total, used the classification of topics/skills from the above seven guidelines.

Initial interviews and synthesizing the topics from literature have been used in 10 and 7 studies, respectively. Eight studies used existing university courses, which carries a heavy risk of overlooking skills/topics required by industry and not addressed by existing university curriculum. Four studies used the SE topics presented in a highly-cited paper by Lethbridge [P33], which is one of the pioneering studies in aligning SE education with industrial needs.

Other SE topic lists have been obtained by mining job advertisements [P16, P26, P35], and using ABET accreditation (Accreditation Board for Engineering and Technology) [67] [P19].

---

[1] The interviewer effect (also called interviewer variance or interviewer error) is the distortion of response to a personal or telephone interview which results from differential reactions to the social style and personality of interviewers or to their presentation of particular questions [58] R. M. Groves, *Survey errors and survey costs.* John Wiley & Sons, 2004..

# of papers

	0	10	20
Existing university courses		8	
SWEBOK v1.0 1999	3		
SWEBOK v2 2004	3		
SWEBOK v3 2014	0		
SEEK 2003	2		
ACM SE 2004	1		
ACM SE 2014	0		
ACM Computer Engineering 2004	1		
ACM IT Curriculum	3		
ACM IS Curriculum	3		
Topics in the Lethbridge paper	4		
Syntheszied from literature	7		
Initial interview		10	
Other	6		
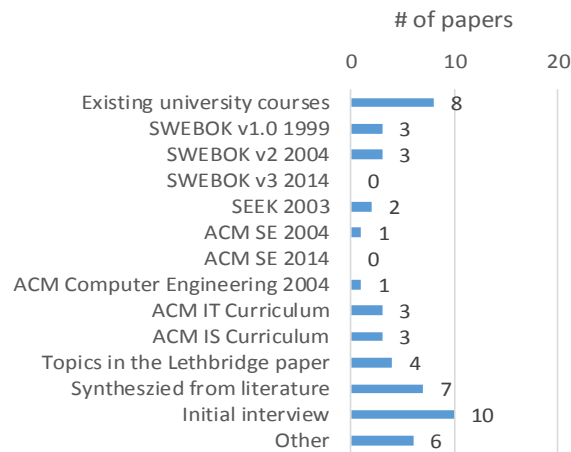
Figure 7- Base models (bodies of knowledge) used for classification of SE topics

6.3 RQ 3: IMPORTANCE OF SE SKILLS

When asking respondents to rate (rank) the importance of SE topics, different papers used different sets of SE topics (as discussed in the previous section) by using different bodies of knowledge. In other words, the questionnaires designed and used in the studies had differences w.r.t. the concrete SE topics used in them.

With such a diversity in the list of SE topics used in the studies, we selected the most relevant model, the latest version of SWEBOK, version 3.0 and mapped the SE topics discussed in the papers to the 15 "knowledge areas" (KAs) of SWEBOK which are: requirements, design (and architecture), development (programming), testing, maintenance, configuration management, project management, SE process, SE models and methods, quality, SE professional practice, SE economics, computing foundations, engineering foundations, and mathematical foundations.

The next step was to consolidate the quantitative data of skill (topic) importance from all the papers. Almost all papers had presented the ranking of the most important skills. To be able to cross-compare and synthesize data in a consolidated manner, we harmonized the importance ranking data as follows. We normalized the ranking of topics in each paper to the range of [0, 1], for each SWEBOK KA. For example, for [P1], three of the 14 ranked topics were related to the "design" KA: general architecture (in rank 1), object-oriented design (rank 9), and user-interface design (rank 12). We calculated the average of (1, 9, 12), which equals 7.33, and divided it by the number of all SE topics in that paper (14), and the normalized rank metric was 0.52. Note that since rank data were used, the lower the value of this metric, the higher the importance of a given topic. Thus, by calculating "1- normalized rank", we aggregated the normalized importance (0.48 for the above example). Once we had the importance metric value of each KA for each paper, we calculated its average among all papers.

Figure 8 shows the most important topics in a scatter plot, in which the normalized importance metrics of each topic and the number of papers that it has appeared in, are shown. Given the fast-changing nature of the field of SE and the knowledge areas in it, we were curious to compare the skill importance data from all the papers, versus the papers published in the last five years. Thus, we calculated the above metrics for each case separately.

Comparison of the two charts in Figure 8 provides interesting insights. When looking at all the papers, requirements, design and testing are the most important and most frequently-mentioned topics with SE professional practice, project management and development coming next. However, when looking at the recent papers, the top-3 topics change to SE professional practice, project management, and testing. This seems to denote that less technical skills such as SE professional practice and project management have become even more important in SE education in recent years. SE professional practice covers topics such as professionalism, group dynamics, and communication skills and these are (soft) skills especially required in modern agile software development that is more strongly based on communication and interaction than traditional waterfall approaches. According to our experience, an effective approach to cover project management and SE professional practice in our education practice is by larger SE projects done by student teams either in class or even together with companies [8].

Mathematical and engineering foundations, as well as SE economics, are ranked low in both charts. This may highlight the establishment of SE (and its education) as a separate engineering discipline that relies on other sciences such as computer science, mathematics and economics and adopting ideas from those subjects by developing new approaches to solve

problems in engineering software [68]. In line with this finding, it is interesting to observe that requirements, testing, and design are considered more important than the actual development. However, according to our experience, this is not always reflected in SE education, especially if it is embedded into computer science curricula.

In addition, it is interesting to observe that there are weak to moderate correlations in the two charts in Figure 8. The Pearson correlation coefficients are 0.63 (top plot) and 0.34 (bottom plot). Thus, it is the case that when a SE topic has appeared in more papers in the pool, the average importance may be higher.
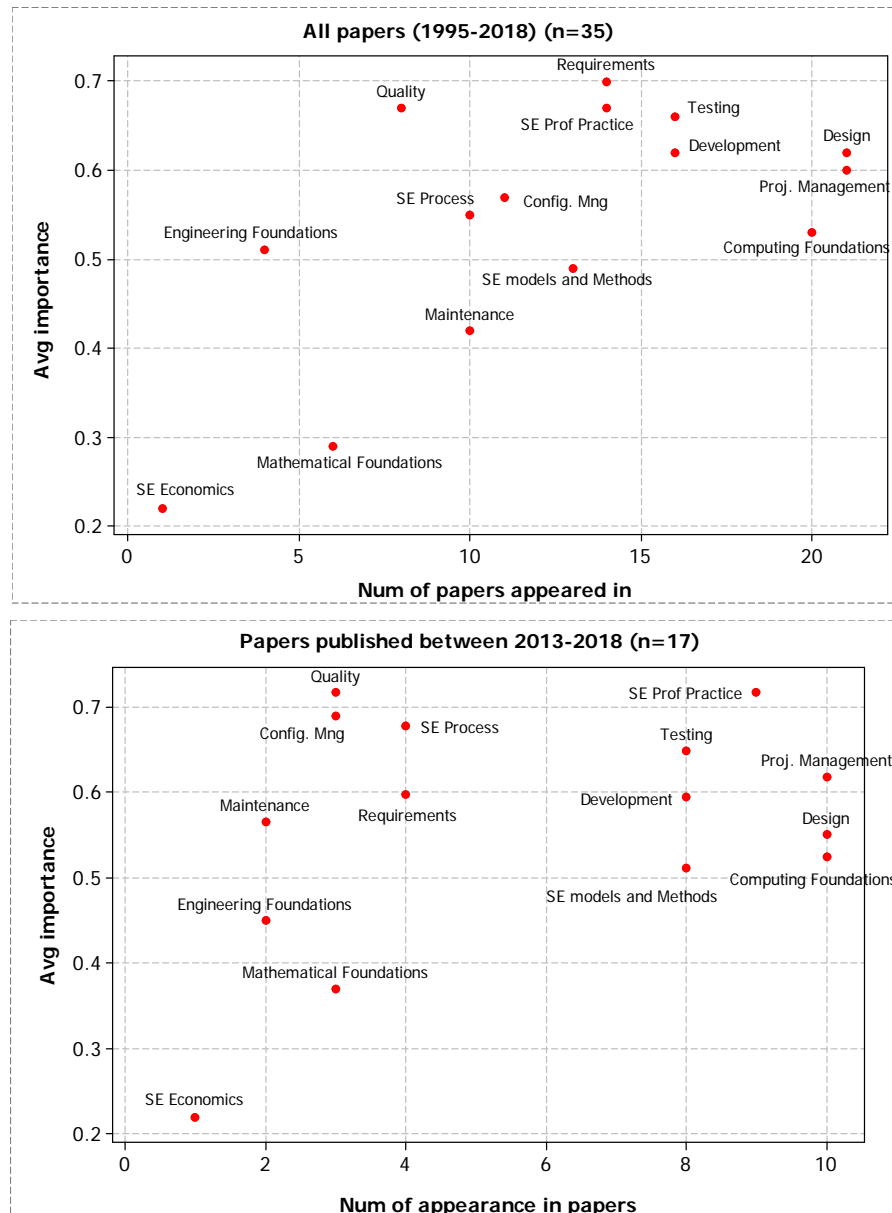
Figure 8- The most important skills (data from all the papers, versus papers in the last five years)

6.4 RQ 4: KNOWLEDGE GAPS (DEFICIENCIES)

Eight (8) of the 35 studies measured, in quantitative terms, the knowledge gap (deficiency) from their survey participants, which was usually done by subtracting the importance-in-job measure of a given SE topic from the measure of how much the participant had learned during her/his university education. We extracted all that quantitative knowledge gap values and calculated their normalized average. We show in Figure 9 two scatter plots to visualize the average knowledge gap values versus their importance. Similar to the previous Figure 8, we show two charts for two cases: (1) all the eight studies reporting knowledge gaps, and (2) six of those eight papers which have been published in the last five years. The X-axis shows the

average importance and the Y-axis the average knowledge gap. As we can see, in the case of all eight papers, generally speaking, the two factors are quite correlated and with increasing reported importance, more knowledge gap has been reported. The greatest reported knowledge gaps are in the following areas: (1) configuration management, (2) SE models and methods, (3) SE process, (4) design (and architecture), and (5) testing. Thus, in general, more education and training focus shall be given to these topics, by university programs and when training newly-hired staff in the industry.

When we look at the recent data (the last five years), the difference w.r.t. both metrics are more widespread and thus it is easier to differentiate and pick the topics with the highest knowledge gaps. We have divided the scatter plots of Figure 9 into four quadrants to be able to clearly see the SE topics with low/high importance and low/high knowledge gap.
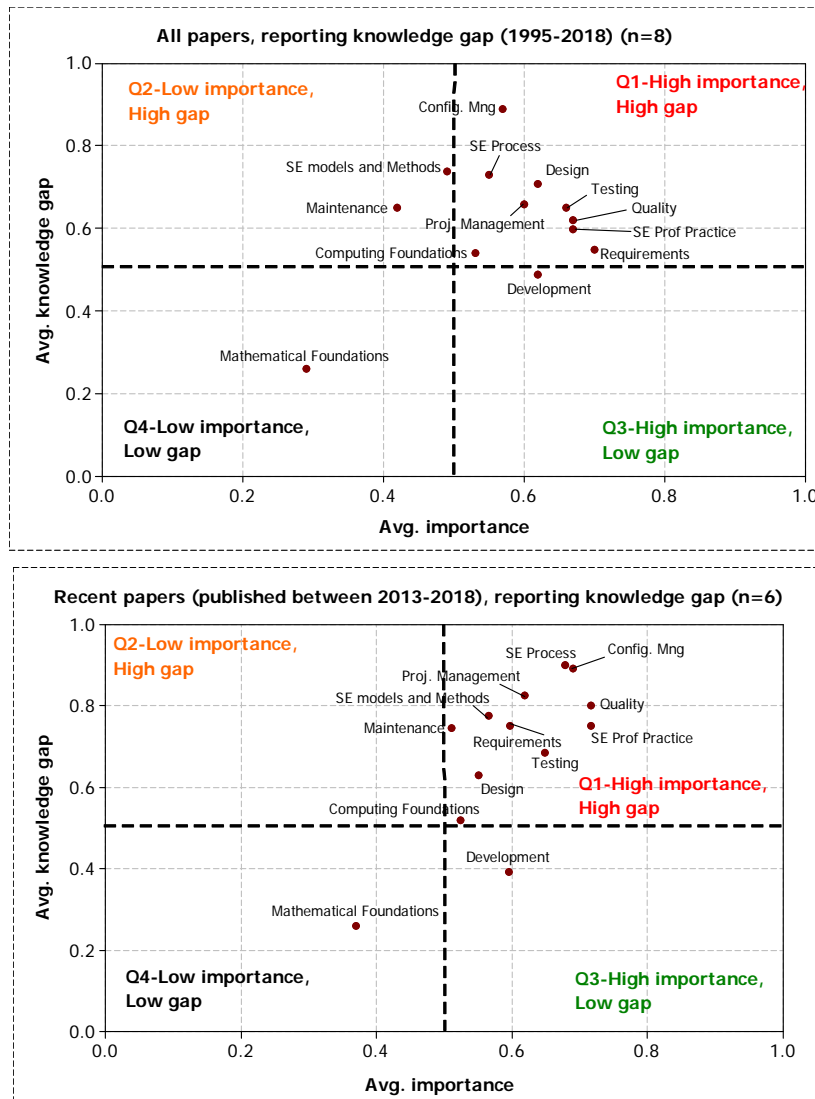
Figure 9- Topics with the greatest knowledge gap— where importance exceeds current knowledge of survey participants

Topics in Q1 (high importance, high gap) are those which need the highest attention w.r.t. need for improvements in SE education in university programs. They have high importance, but high knowledge gap. Topics in Q2 (low importance, high gap) should get the next level of attention w.r.t. SE education (after those in Q1). They have relatively low importance, but there are still high knowledge gaps in those topics, and thus need for more education and training on those topics.

In terms of topics in Q3 (high importance, low gap), the university programs are generally doing a good job, since knowledge gaps in those topics are relatively low, while they are quite important w.r.t technical needs in the industry. Only the software "development" topic slightly falls in Q3 in one of the scatter plots in Figure 9.

15

Topics in Q4 have low importance, and low knowledge gap, thus they are the least critical w.r.t. need for improvements and attention of SE education in university programs. The KA "Mathematical foundations" falls into Q4 in both scatter plots.

6.5 RQ 5: IMPORTANCE OF SOFT SKILLS

It is widely discussed in the community that hard (technical) skills alone do not make a great software engineer [69] and soft skills are equally important (if not more). Hard skills are comprised of domain knowledge and technical skills, while soft skills are comprised of the team and interpersonal skills. "*Soft skills contribute significantly to individual learning, team performance, client relations and awareness of the business context*" [P16].

As discussed in Section 5.3, to synthesize data for this RQ, we used qualitative coding [54]. While extracting the raw data from the primary studies (soft skills deemed important), we noticed that a large variety of soft skill phrases were mentioned in the papers, and presenting them all in our SLR would, in their "raw" form, lead to an outcome which could look raw and un-synthesized. By following qualitative coding [54], groups of soft skills emerged as we synthesized the data.

As a result of qualitative coding, we categorized the soft skills as shown in Figure 10: Teamwork and communication (discussed in 21 studies), Leadership (14 studies), Critical thinking (11 studies) and Others (19 studies). Many other important soft skills have been mentioned in the papers, such as cultural fit, understand business drives, aptitude, attitude, coping with ambiguity, learning and curiosity, and passion/drive to innovate.

Some studies even reported quite bold findings, e.g., survey data of an American study [P9] showed that "*soft skills are significantly more important than hard skills for entry-level positions*". A study performed in New Zealand [P5] reported that: "*Soft skills are critical skills in SE and make up seven of the top eight most important skills [in that study]*". [P23] also recommended that: "*Soft skills and business skills must be included in curricula*".

These statements are in line with the finding that SE professional practice is of high importance (see Figure 8) which comprises of topics such as professionalism, group dynamics, and communication skills.
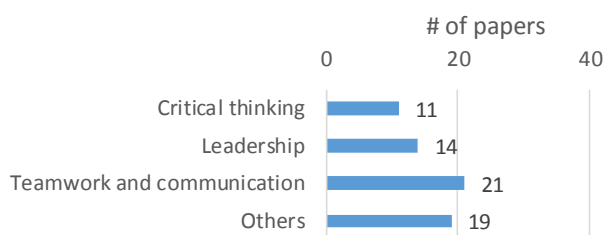
Figure 10- Importance of soft skills as discussed in the papers

6.6 RQ 6: DATA SAMPLE CHARACTERISTICS

In terms of the number of data points (respondents of surveys), studies had between eight [P28] and 600 respondents [P21]. We show in Figure 11 the histogram of the number of data points in each study. Three studies had participants between 543 and 600 [P16, P21, P22], whereas the rest had less than 300 data points.

Since the studies were mostly conducted in different countries, there is a small chance that a single software engineer had participated in more than one study in the pool. Thus, when we add up the number of respondents from all 35 studies, we can say that the data and evidence are from "up to" 4,132 respondents. By combining data and evidence from all previous studies and by having such a large combined dataset, our study aims to provide a more comprehensive view of the topic.
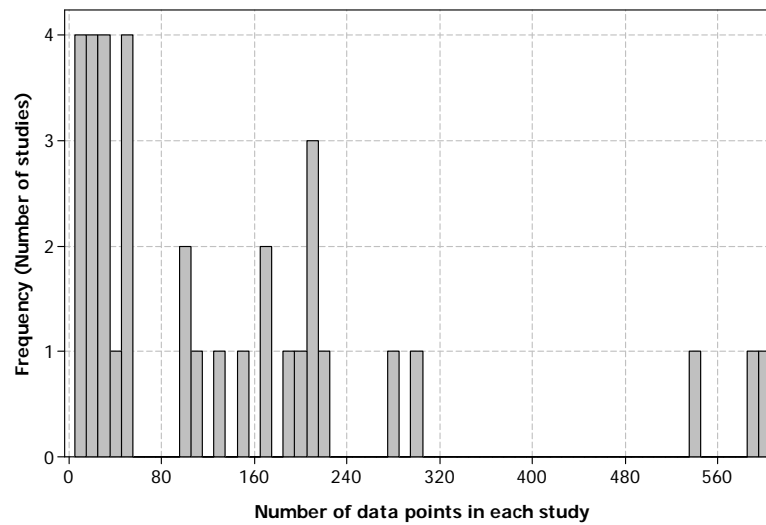
Figure 11- Histogram of the number of data points in each study

Furthermore, we show in Table 3 the list of countries where the data have been gathered from and also the study references. The top countries from which data were gathered were: USA (16 papers), Canada (4), South Africa (4), New Zealand (2) and Spain (2). Each of the following countries was represented in one paper each: England, Norway, Philippines, Jordan, Australia, Finland and Samoa. Two papers had data from both the USA and Canada [P31, P33] and one paper surveyed data from all over the world [P25]. The advantage of our meta-synthesis (meta-analysis) is that the combined dataset has data from 13 countries which provides a stronger evidence on the subject.

At the same time, Table 3 shows a strong bias of the combined (synthesized) data-set towards North America. More than half of the studies (20 out of 35) have been conducted in the US and Canada, respectively, which means the results might not be applicable in other countries, i.e., in particular some countries in Europe but also India and China are missing. Thus, the results of the study should be interpreted carefully and treated with "a grain of salt". We have added a discussion in the threats to validity, about this issue.

Table 3: List of countries w.r.t. where the data have been gathered from

Countries	Num. of studies	Study references
USA	16	[P1, P7, P8, P9, P10, P11, P13, P15, P18, P19, P20, P21, P22, P24, P29, P33]
Canada	4	[P1, P31, P32, P33]
South Africa	4	[P17, P23, P28, P30]
New Zealand	2	[P5, P16]
Spain	2	[P6, P14]
Samoa	1	[P2]
England	1	[P3]
Norway	1	[P4]
Philippines	1	[P12]
Jordan	1	[P26]
Netherlands	1	[P35]
Australia	1	[P27]
Finland	1	[P34]
Worldwide	1	[P25]

6.7 RQ 7: SYNTHESIS OF EDUCATIONAL RECOMMENDATIONS

To answer RQ 7, we looked for common and interrelated themes to build a holistic view on aligning SE education with industry needs. To achieve this, we used qualitative coding [54].

Application of qualitative coding provided four themes for the educational recommendations provided in the papers, as shown in Figure 12: (1) Need for more emphasis on soft skills (20 papers), (2) Need for active IAC (3 papers), (3) Less emphasis on certain topics (2 papers), and (4) Other recommendations (7 papers). Note that we intentionally did not create

17

a group for "More emphasis on certain topics" since such recommendations are obviously provided in all the papers, given the scope of our review.

# of papers

| | 0 | 20 | 40 |

Need for more emphasis on soft skills 20
Need for active IAC 3
Less emphasis on certain topics 2
Other recommendations 7
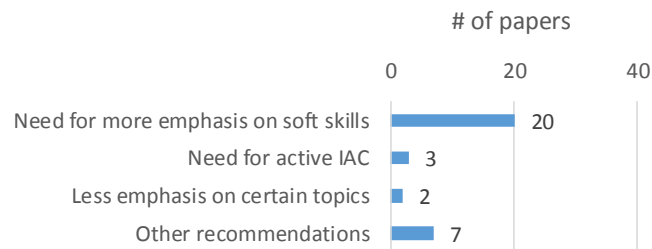
Figure 12- Qualitative coding of educational recommendations, presented in the papers

Soft skills are important for software engineers since SE activities are carried out in a social context that is made up of people with different personalities. As being a part of such a social context, software engineers should possess some soft skills to fulfill their responsibilities. 20 of the 35 studies emphasized the importance of soft skills and identified a need for improving students' soft skills during university education. [P7] and [P9] reported that soft skills are even more valued by industry than hard skills. The rationale behind this is that soft skills are not directly teachable and requires extensive effort and time to improve compared to hard skills [70]. [P7] and [P22] also pointed out the increasing importance of soft skills in parallel with an increase in the level of position in an organization. Hard skills are more important for entry-level positions [P7, P22], whereas soft skills are becoming more important at higher levels. Such a recommendation has also been supported by studies outside SE, e.g., [71], which mentioned that, in higher position levels, managers' effectiveness depends largely on soft skills.

To provide SE students a setting for improving soft skills as well as applying hard skills, it is suggested to use real-life projects, e.g., [P20, P23, P26, P27, P28], which emphasized the importance of internships and real-life projects as a part of formal SE education. [P28] reported that "*Real-life and practical experience must be included in students' education*". [P27] also highlighted the need for "*more exposure to real life, exercises, team assignments or industry projects*". Such complementary activities enable students to participate in the learning process actively and hence provide them with an environment for improving soft and hard skills [P27]. On the other hand, these benefits do not come for free. Instructors should envisage the effort for planning, running, and coordinating such activities [P26]. Moreover, when a part or all of these activities need the participation of stakeholders from industry, a proper setting should be established for industry-academia collaboration [26]. Besides providing real-life projects, the industry can also support academia in developing curriculum [P12, P28] and also in delivering some lectures [P26].

Another important point to be raised is keeping SE curriculum continually up-to-date [72]. As Boehm stated, it is important to "*anticipate future trends, prepare students to deal with them, monitor current principles and practices and separate timeless principles from out-of-date practices*" [72]. In parallel with this, [P1, P28] emphasized the need for curriculum change based on inputs from industry. Aligned with curriculum updates, lecturers/educators should also adapt themselves to the changes in the SE field [P30].

We also observed many other interesting findings when reviewing the papers. For example, there were suggestions on decreasing emphasis on certain topics in SE university education (i.e., what we should teach less). [P1] expressed that "*Participants felt that their university education gave them a much better grounding in mathematics than in software topics*", and thus recommended that: "*emphasis on certain mathematics topics should be changed [decreased]*". The empirical data also showed that "*much mathematics is being forgotten, whereas much new software knowledge is being acquired on-the-job*". [P3] also reported that there is "*over-emphasis on mathematical topics and under-emphasis on business topics*" in SE education. [P3] called for a less educational focus on parsing and compiler design, formal specification methods, digital electronics and digital logic in SE programs.

Several interesting suggestions were also made in [P29], e.g., about benefits of working with existing code-bases: "*Instead of a greenfield project, a more valuable experience would provide students a large pre-existing codebase to which they must fix bugs (injected or real) and write additional features".* [P29] also suggested to include training about management skills as it said: "*Also valuable would be a management component, where students must interact with more experienced colleagues (students who have taken the class previously, who can act as mentors) or project managers (teaching assistants) who teach them about the codebase*".

6.8 RQ 8: COMPARING OUR RESULTS TO PREVIOUS REVIEW PAPERS

As discussed in Section 2, the RQs of three related survey papers [1, 22, 23] have similarities to our RQs. We provide a comparison of findings w.r.t. to those RQs from those studies with our findings in Table 4.

In [23], the authors performed a meta-analysis of the 155 papers in the ICT area. The RQs of that study were broader and concentrated on clustering the range of topics as follows: curriculum design and delivery, mismatch between learning outcomes of ICT graduates and skills requirements from the labor market, different/new teaching assessment methods in the education of ICT professionals, improvement in collaboration between ICT academics and ICT practitioners, and employability of ICT graduates. Based on these clusters, the authors compiled a list of educational recommendations from the analyzed papers. According to the study, while a considerable number of papers describe challenges and bottlenecks in education and careers of ICT professionals; only a few papers identify and justify new approaches for proposing solutions to challenges in ICT education and ICT career development. The main recommendation of the paper was to position student employability as the key metric for universities.

Table 4-Comparison with other related literature review papers

Criteria	Related literature review paper			
	[1]	**[22]**	**[23]**	**This study**
Literature review period	1995-2012	1995-2013	1980-2014	1995-2018
Top knowledge deficiencies identified (technical)	Testing, programming, project management	Design, testing, configuration management tools	- (not reported)	Configuration management, SE models and methods, design
Top knowledge deficiencies identified (soft skills)	Teamwork, oral communication, problem solving	Oral and written communication teamwork ethics	-	Critical thinking, leadership, teamwork, and communication
Ranking mechanism of knowledge deficiencies	Counting the number of occurrences of the deficiencies	Counting the number of occurrences of the deficiencies	-	Relative ranking provided by the papers
Change in knowledge deficiencies over time	Trends were provided per each knowledge area	-	-	Overall and last 5-years results were compared
Taxonomy used for classification of knowledge deficiencies	Authors' own classification	Authors' own classification	-	SWEBOK v3
Importance of skills	-	-	-	Importance of skills is provided if found in the primary paper.
Synthesis of educational recommendations	-	-	List of educational recommendations classified into key areas	List of educational recommendations classified according to different stakeholders
Description of data sample characteristics (as discussed in our RQ 6)	-	Not explicitly provided	Not explicitly provided	Explicitly provided
Targeted industry role	Software engineers/ developers	Software engineers/ developers	ICT professionals	Software engineers/ developers

Another study [22] presented a literature review on exploring the areas where graduating students most frequently fall short of the expectations of industry. The main findings of the results indicated that the graduating students are lacking in both technical abilities (design, testing, configuration management tools etc.) and personal skills (oral and written communication, teamwork, etc.). The authors analyzed the trends of knowledge deficiencies of students over time. The authors counted the number of occurrences where a knowledge deficiency is mentioned in the primary studies rather than ranking them.

The other study [1] is a Master's thesis by the first author of [22]. By being slightly different than [22], the study provided the trends of knowledge deficiencies over the years which could be potentially valuable. The other significant difference is that the study provided an overview of knowledge deficiencies across different perspectives (managers, new hires, experienced employees, and students). It was found worthwhile to compare the viewpoints of different roles on this issue.

7 DISCUSSION

We discuss next some of the possible benefits of this review (Section 7.1) and then we discuss potential threats to validity of this review (Section 7.2) and the steps that we have taken to mitigate or minimize them.

7.1 POTENTIAL BENEFITS OF THIS REVIEW

We discuss next the potential benefits of our SLR for educators, researchers, hiring managers and students.

Benefits for educators and researchers:

- Adapting university curricula based on industry's needs: The information we have provided in RQ 3 (Importance of SE skills), RQ 4 (Knowledge gaps) and RQ7 (synthesis of educational requirements) could be used by educators to address the knowledge deficiencies in SE and improve their curricula.
    - o For example, among our findings was that, when looking at the recent papers, the top-3 most important SE skills are SE professional practice, project management, and testing. This seems to denote that skills such as SE professional practice and project management (which used to be less important) have become more important in industry in recent years. SE professional practice covers topics such as professionalism, group dynamics, and communication skills and these are (soft) skills especially required in modern agile software development.
    - o We found that the greatest knowledge gaps are in the following areas: (1) configuration management, (2) SE models and methods, (3) SE process, (4) design (and architecture), and (5) testing. Thus, in general, more education and training focus shall be given to these topics, by university programs and SE educators.

- Future education research on the topic: This SLR can be a valuable resource for future research on knowledge deficiencies and their importance by providing researchers with a structured roadmap. We provide our concrete suggestions below:
    - o Furthermore, while we developed a suitable approach (see Section 5.2) to extract and then synthesize the relative ranking data from the primary studies, it would be a good idea to have an established survey "protocol" (perhaps based on SWEBOK) which future primary studies (opinion surveys about knowledge gaps) could use to replicate surveys. When (and if) established and used by many future survey studies, such a "uniform" survey protocol could facilitate cross comparisons and future meta-analysis studies like this work.
    - o We also think future education research could look into the "challenges" of aligning SE education with industrial needs. While perhaps many SE educators are possibly aware of SE areas with high knowledge gaps in industry, what are the issues in curriculum and university contexts which make it challenging to better align SE education w.r.t. those knowledge gaps?
    - o We saw in Section 6.6 that about half of the studies extracted data in the US (16 of 35 papers). Also several other "clusters" of studies were reported for a few other countries, such as Canada. Further research could conduct meta-analysis on such sub-datasets only, to provide regional insights.

Benefits for hiring managers:

- Addressing the knowledge deficiencies: The reported knowledge deficiencies from the literature that we have summarized in RQ3, RQ4, and RQ5 could potentially be used by the industry when designing an onboarding program for new hires. For example, the proposed onboard training program would focus on certain topics to address the knowledge deficiencies. In the context of a large software company in Turkey, with which one of the authors was affiliated with, an industrial training program for potential new hires was recently conducted [12] based on the insights provided by this review study.
- Calibrating the expectations and interview process of new hires: This study provides an overview of expected capabilities of new hire student population in different dimensions. Based on this data, the hiring manager would have a general expectation of a new hire and could potentially change their expectations for a new hire and the interview process.
- Assessing employee competencies and deciding for new hires: The organization should support the current project groups with the missing expertise and therefore, the output of this study can help managers to observe the neglected roles in the organization and find new hires for these missing competencies required.

Benefits for students:

- Addressing their knowledge gaps: The reported knowledge deficiencies from the literature that we have summarized in RQ3, RQ4, and RQ5 could potentially be used by the students to address the gap by themselves. This reported knowledge deficiencies could be addressed via different methods such as MOOCs (Massive Open Online Courses), certificate programs, self-study, and internships. Students can learn the important skills required for the software industry before applying for and starting SE positions. While doing a side project that the student is passionate about, the student can improve her/his skills by considering the skills emphasized in this study.

7.2 POTENTIAL THREATS TO VALIDITY

We carefully identified and addressed potential threats to validity in each step of the study. We followed the guidelines for performing systematic literature reviews and benefited from our previous experience, e.g., [26, 27]. This sub-section discusses the potential threats to validity and the actions we took to minimize or mitigate them.

Internal validity: A systematic approach has been applied to conduct this study, as described in Section 3, 4, and 5. To enable the repeatability of this study, we have defined and reported all the steps, including search engines, search terms, inclusion/exclusion criteria used. Some problems related to internal validity might have occurred due to the limitation of search engines and terms and bias in applying inclusion/exclusion criteria.

To have a finite set of primary studies for review, it is mandatory to limit the search engines and terms used. Such a limitation is a threat for the completeness. To decrease the possible negative impact of this risk, all the authors conducted independent searches using the search terms. In addition, we also conducted forward and backward snowballing [29] on the set of papers already in the pool.

To minimize the risk of bias in applying inclusion/exclusion criteria, authors voted for the papers to be included. The papers which had conflicting votes have been explicitly discussed by the authors. A consensus has been reached for all of such papers.

Construct validity: The suitability of RQs and categorization scheme used for data extraction address the construct validity for this study. We believe that with a suitable list of RQs and a good design of the categorization scheme in the way that it would better relate to the theoretical concepts behind this study, this study would have higher construct validity.

We thoroughly examined the related work and extracted RQs and a categorization scheme based on the current literature. The RQs cover our research goals, which have been answered according to the categorization scheme. Moreover, peer reviews were performed to increase the quality of data extracted.

Conclusion validity: To be able to provide scientific conclusions, it is mandatory to treat the primary studies rigorously and in a repeatable manner. We treated all the papers we could have reached through a carefully designed and executed search process. The authors refined the categorization scheme iteratively to avoid any ambiguity. Peer review has mitigated the risk of bias in data extraction and disagreements have been resolved by consensus. Our well-defined, executed and reported research method ensures the replicability of this study without major differences in the conclusions.

Another issue w.r.t. our meta-analysis is due to the data gathered in primary studies, which applied different data collection methods. As discussed in Section 6.1, most studies have used opinion surveys, while some have used interviews, or data from mining job advertisements. According to the guidelines for meta-analysis in SE [7, 55, 56], we did not see any major issue when aggregating data from these multiple sources. It is actually common in Empirical SE to gather data from different sources and with different data collection methods [7, 55, 56], and it would be a form of multi-method (or mixed method) approach.

Related to the above possible issue, is the fact that different primary studies have used different base knowledge models (as discussed in Section 6.2). As discussed in Section 6.3, we harmonized the ranking data for SE skills w.r.t. different knowledge models by classifying the SE topics in each of the primary studies under knowledge areas of SWEBOK. We believe this data harmonization approach was a sound and reasonable practice and we observed that the classification went smoothly.

Let us recall from Section 6.6 that our meta-analysis aggregated the data from a set of primary studies stemming from 13 countries, and we are aware that, we in no way assume a similar reality for the software industry and the SE education in different countries. In any country, the software industry needs and the SE education are influenced by a set of context variables (usually related to the local reality) such as maturity level of the software industry (including the stakeholders), type of software products developed in the region (e.g., embedded systems or games), socio-economic conditions, level of IT adoption, etc. Let us also recall from Section 6.6 that when we added up the number of respondents from all 35 studies, the combined dataset included 4,132 data points (respondents). Thus, our meta-analysis is similar to a large-scale survey which would have received that many data points from the 13 countries listed, and we know that many such opinion survey studies (covering many countries) are regularly reported in SE and their findings are considered valuable. Furthermore, about half of the studies collected data in the US (16 of 35 papers). Thus, further research could conduct meta-analysis of such datasets only, to provide regional insights.

External validity: External validity is defined as to what extent the results of this study can be generalized. We extracted data from all studies in the relevant literature, which provided sufficient information.

However, as discussed in Section 6.6, we observed a strong bias of the combined (synthesized) data-set towards North America. We saw that more than half of the studies (20 out of 35) have been conducted in the US and in Canada, which means the results might not be applicable in other countries, i.e., in particular some European countries and also India and China are missing. Thus, the results of the study should be interpreted careful and treated with "a grain of salt". In addition, the focus of the studies lies on traditional universities, which have a quite similar educational canon all over the world. However, the generalizability to other tertiary education institutes like colleges, polytechnic universities or universities of applied sciences may be limited.

8 CONCLUSIONS AND FUTURE WORK

The findings presented in this article show the importance of SE professional practice and soft skills in general, the importance of certain areas in SE education (especially requirements, design and testing), knowledge gaps in specific areas of SE (especially configuration management, SE models and methods as well as SE process), and the importance of real-world examples in SE courses.

The authors have already started to benefit from the findings of the presented review and meta-analysis study in their SE education activities in several ways. This review has helped us to identify the most important SE topics, based on the largest synthesized body of evidence in the literature. Also, we found that the greatest knowledge gaps are in configuration management, SE models and methods, SE process, design (and architecture), and testing. Furthermore, in our ongoing university SE courses, we have started to align our teaching materials with the important topics and areas which have the greatest knowledge gaps. Also in the context of a large software company in Turkey, with which one of the authors was affiliated with, an industrial training program for potential new hires was recently conducted [12] based on the insights provided by this review study. We are certain that the results and findings presented in this paper will also benefit other educators and hiring managers by helping them adapt their education/hiring efforts to best prepare the SE workforce.

Finally, the findings also show that mathematical and engineering foundations are often overemphasized in SE programs. This highlights the need to further establish SE as a separate engineering discipline using knowledge from computer science and other basic sciences such as mathematics, economics or even psychology, and to further separate computer science from SE university programs [73]. This literature review can serve as a starting point for future research into knowledge deficiencies and their importance by proving researchers with a structured roadmap, and existing knowledge deficiencies.

Our future work will include (1) performing a recent survey on this topic; (2) assessing a set of SE programs in several universities based on the results of this study; and (3) conducting interviews with software companies to get their feedback about our empirical results.

9 REFERENCES

This section is divided into two parts: (1) Citations to the primary studies reviewed in the SLR; and (2) Other (regular) references cited throughout the paper.

9.1 PRIMARY STUDIES (SOURCES REVIEWED IN THE SLR)

[P1] T. C. Lethbridge, "A survey of the relevance of computer science and software engineering education," in Conference on Software Engineering Education, 1998, pp. 56-66.

[P2] I. C. Mow, H. Sasa, F. Maua-Faamau, E. Mauai, M. Tanielu, "An Evaluation of Relevance of Computing curricula to Industry Needs", in Systemics, Cybernetics and Informatics, vol. 13, no. 1, pp. 7-12, 2015.

[P3] B. Kitchenham, D. Budgen, P. Brereton, P. Woodall, "An investigation of software engineering curricula," Journal of Systems and Software, vol. 74, no. 3, pp. 325-335, 2005.

[P4] A. Deak, G. Sindre, "Analyzing the importance of teaching about testing from alumni survey data," Norsk informatikkonferanse (NIK), 2013.

[P5] C. Watson, K. Blincoe, "Attitudes towards software engineering education in the New Zealand industry," in 28th Annual Conference of the Australasian Association for Engineering Education (AAEE 2017), 2017, pp. 785-792.

[P6] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F. J. García-Peñalvo, E. Tovar-Caro, "Competence gaps in software personnel: A multi-organizational study," Computers in Human Behavior, vol. 29, no. 2, pp. 456-461, 2013.

[P7] M. E. McMurtrey, J. P. Downey, S. M. Zeltmann, W. H. Friedman, "Critical skill sets of entry-level IT professionals: An empirical examination of perceptions from field personnel," Journal of Information Technology Education: Research, vol. 7, pp. 101-120, 2008.

[P8] D. M. Lee, E. M. Trauth, D. Farwell, "Critical skills and knowledge requirements of IS professionals: a joint academic/industry investigation," MIS quarterly, pp. 313-340, 1995.

[P9] K. Jones, L. N. Leonard, G. Lang, "Desired Skills for Entry Level IS Positions: Identification and Assessment," Journal of Computer Information Systems, vol. 58, no. 3, pp. 214-220, 2018.

This is the pre-print of a paper that has been published in the Journal of Systems and Software:
https://doi.org/10.1016/j.jss.2019.06.044

[P10]   R. D. Howard, "Does the information systems curriculum meet business needs: Case study of a southeastern college," PhD thesis, Capella University, 2017.

[P11]   C. Scaffidi, "Employers' need for computer science, information technology and software engineering skills among new graduates," International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), vol. 8, no. 1, pp. 1-12, 2018.

[P12]   F. Patacsil, C. L. S. Tablatin, "Exploring the importance of soft and hard skills as perceived by IT internship students and industry: A gap analysis," JOTSE, vol. 7, no. 3, pp. 347-368, 2017.

[P13]   A. Radermacher, "Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectations of industry," PhD thesis, North Dakota State University, 2012.

[P14]   R. Colomo-Palacios, E. Tovar-Caro, Á. García-Crespo, J. M. Gómez-Berbís, "Identifying technical competences of IT Professionals: the case of software engineers," International Journal of Human Capital and Information Technology Professionals (IJHCITP), vol. 1, no. 1, pp. 31-43, 2010.

[P15]   G. Lang, K. Jones, L. N. Leonard, "In the know: Desired skills for entry-level systems analyst positions," Issues in Information Systems, vol. 16, no. 1, pp. 142-148, 2015.

[P16]   M. Stevens, R. Norman, "Industry expectations of soft skills in IT graduates: a regional survey," in Proceedings of the Australasian Computer Science Week Multiconference, ACM, 2016, pp.13-21.

[P17]   J. Liebenberg, M. Huisman, E. Mentz, "Industry's perception of the relevance of software development education," TD: The Journal for Transdisciplinary Research in Southern Africa, vol. 11, no. 3, pp. 260-284, 2015.

[P18]   A. Radermacher, G. Walia, D. Knudson, "Investigating the skill gap between graduating students and industry expectations," in Companion Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 291-300.

[P19]   C. L.Aasheim, L. Li, J. D. Shropshire, C. A. Kadlec, "IT program curriculum recommendations based on a survey of knowledge and skill requirements for entry-level IT workers," Information Technology Faculty Publications, Paper 11, 2011.

[P20]   C. V. Bullen, T. Abraham, K. Gallagher, J. C. Simon, P. Zwieg, "IT workforce trends: Implications for curriculum and hiring," Communications of the association for information systems, vol. 20, pp. 545-554, 2007.

[P21]   C. L. Aasheim, S. R. Williams, "Knowledge and skill requirements for entry-level information technology workers: do employers in the IT industry view these differently than employers in other industries?," Journal of Information Systems Education, vol. 20, no. 3, pp. 349-356, 2009.

[P22]   C. L. Aasheim, S. Williams, E. S. Butler, "Knowledge and skill requirements for IT graduates," Journal of Computer Information Systems, vol. 49, no. 3, pp. 48-53, 2009.

[P23]   J. Liebenberg, M. Huisman, E. Mentz, "Knowledge and Skills Requirements for Software Developer Students," International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, vol. 8, no. 8, pp. 2604-2609, 2014.

[P24]   A. Radermacher, G. Walia, D. Knudson, "Missed expectations: where CS students fall short in the software industry," CrossTalk: The Journal of Defense Software Engineering, vol. 28, no. 1, pp. 4-8, 2015.

[P25]   T. C. Lethbridge, "Priorities for the education and training of software engineers," Journal of Systems and Software, vol. 53, no. 1, pp. 53-71, 2000.

[P26]   S. Hanna, H. Jaber, A. Almasalmeh, F. A. Jaber, "Reducing the gap between software engineering curricula and software industry in Jordan," Journal of Software Engineering and Applications, vol. 7, pp. 602-616, 2014.

[P27]   O. Minor, J. Armarego, "Requirements Engineering: a close look at industry needs and a model curricula," Australasian Journal of Information Systems, vol. 13, no. 1, pp. 192-208, 2005.

[P28]   J. Liebenberg, M. Huisman, E. Mentz, "Software: university courses versus workplace practice," Industry and Higher Education, vol. 29, no. 3, pp. 221-235, 2015.

[P29]   A. Begel, B. Simon, "Struggles of new college graduates in their first software development job," In ACM SIGCSE Bulletin, vol. 40, no. 1, pp. 226-230, 2008.

[P30]   J. Liebenberg, M. Huisman, E. Mentz, "The relevance of software development education for students," IEEE Transactions on Education, vol. 58, no. 4, pp. 242-248, 2015.

[P31]   T. C. Lethbridge, "The relevance of software education: A survey and some recommendations," Annals of Software Engineering, vol. 6, no. 1-4, pp. 91-110, 1998.

[P32]   A. Seffah, "Training developers in critical skills," IEEE software, vol. 16, no. 3, pp. 66-70, 1999.

[P33]   T. C. Lethbridge, "What knowledge is important to a software professional?," Computer, vol. 33, no. 5, pp. 44-50, 2000.

[P34]   S. Surakka, "What subjects and skills are important for software developers?," Communications of the ACM, vol. 50, no. 1, pp. 73-78, 2007.

[P35]   M. Daneva, C. Wang, P. Hoener, "What the job market wants from requirements engineers?: an empirical analysis of online job ads from the netherlands," in Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2017, pp. 448-453.

9.2 BIBLIOGRAPHY

[1]   A. D. Radermacher, "Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectations of industry," MSc thesis, North Dakota State University, 2012.

This is the pre-print of a paper that has been published in the Journal of Systems and Software:
https://doi.org/10.1016/j.jss.2019.06.044

[2]   I. Hire, "Mind the Gap: A Report on the UK's Technology Skills Landscape," *www.hired.com/skills-gap*, Last accessed: Feb. 2019.

[3]   G. Ford and N. E. Gibbs, "A master of software engineering curriculum," *IEEE Computer,* vol. 22, no. 9, pp. 59-71, 1989.

[4]   J. Baker, "2018's Software Engineering Talent Shortage— It's quality, not just quantity," *https://hackernoon.com/2018s-software-engineering-talent-shortage-its-quality-not-just-quantity-6bdfa366b899*, Last accessed: Feb. 2019.

[5]   American Society for Training Development, "Bridging the skills gap: help wanted, skills lacking: why the mismatch in today's economy?," *Technical report, https://www.nist.gov/sites/default/files/documents/mep/Bridging-the-Skills-Gap_2012.pdf*, 2012.

[6]   K. Beckman, N. Coulter, S. Khajenoori, and N. R. Mead, "Collaborations: closing the industry-academia gap," *IEEE software,* vol. 14, no. 6, pp. 49-57, 1997.

[7]   D. S. Cruzes and T. Dyba, "Synthesizing evidence in software engineering research," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2010, pp. 1-10, 1852788.

[8]   V. Garousi, "Incorporating Real-World Industrial Testing Projects in Software Testing Courses: Opportunities, Challenges, and Lessons Learned," in *Proceedings of the IEEE Conference on Software Engineering Education and Training (CSEE&T)*, 2011, pp. 396-400.

[9]   V. Garousi and A. Mathur, "Current State of the Software Testing Education in North American Academia and Some Recommendations for the New Educators " in *Proceedings of IEEE Conference on Software Engineering Education and Training*, 2010, pp. 89-96.

[10]   V. Garousi, "An Open Modern Software Testing Laboratory Courseware: An Experience Report " in *Proceedings of the IEEE Conference on Software Engineering Education and Training*, 2010, pp. 177-184.

[11]   V. Garousi, "Applying Peer Reviews in Software Engineering Education: An Experiment and Lessons Learned," *IEEE Transactions on Education,* vol. 53, no. 2, pp. 182-193, 2010.

[12]   E. Tuzun, H. Erdogmus, and I. G. Ozbilgin, "Are computer science and engineering graduates ready for the software industry?: experiences from an industrial student training program," in *Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training*, 2018, pp. 68-77.

[13]   G. Giray, E. Tüzün, and V. Garousi, "Assessment of the Turkish software engineering university programs using the SWEBOK guide," in *Proceedings of the Turkish National Software Engineering Symposium (in Turkish)*, 2016, pp. 574-585.

[14]   V. Garousi, M. M. Eskandar, and K. Herkiloğlu, "Industry-academia collaborations in software testing: experience and success stories from Canada and Turkey," *Software Quality Journal,* vol. 25, no. 4, pp. 1091–1143, 2017.

[15]   D. S. Cruzes, M. Felderer, T. D. Oyetoyan, M. Gander, and I. Pekaric, "How is Security Testing Done in Agile Teams? A Cross-Case Analysis of Four Software Teams," in *Agile Processes in Software Engineering and Extreme Programming*, 2017, pp. 201-216.

[16]   B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software engineering," in "EBSE Technical Report," 2007, vol. EBSE-2007-01.

[17]   A. X. Garg, D. Hackam, and M. Tonelli, "Systematic review and meta-analysis: when one study is just not enough," *Clinical Journal of the American Society of Nephrology,* vol. 3, no. 1, pp. 253-260, 2008.

[18]   B. Malik and S. Zafar, "A systematic mapping study on software engineering education," in *Proceedings of World Academy of Science, Engineering and Technology*, 2012, vol. 6, no. 11, pp. 3343-3353.

[19]   A. Heredia, R. C. Palacios, and A. de Amescua Seco, "A Systematic Mapping Study on Software Process Education," in *Proceedings of the International Workshop on Software Process Education, Training and Professionalism*, 2015, pp. 7-17.

[20]   M. R. Marques, A. Quispe, and S. F. Ochoa, "A systematic mapping study on practical approaches to teaching software engineering," in *IEEE Frontiers in Education Conference*, 2014, pp. 1-8.

[21]   S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, "Requirements engineering education: a systematic mapping study," *Requirements Engineering,* vol. 20, no. 2, pp. 119-138, 2015.

[22]   A. Radermacher and G. Walia, "Gaps between industry expectations and the abilities of graduates," in *Proceeding of ACM technical symposium on Computer science education*, 2013, pp. 525-530.

[23]   K. P. Aničić, B. Divjak, and K. Arbanas, "Preparing ICT Graduates for Real-World Challenges: Results of a Meta-Analysis," *IEEE Transactions on Education,* vol. 60, no. 3, pp. 191-197, 2017.

[24]   V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, "Closing the gap between software engineering education and industrial needs," *IEEE Software, In press, Pre-print: https://arxiv.org/abs/1812.01954*, 2019.

[25]   P. Bourque and R. E. Fairley, "Guide to the Software Engineering Body of Knowledge (SWEBOK), version 3.0," *IEEE Computer Society Press,* 2014.

[26]   V. Garousi, K. Petersen, and B. Özkan, "Challenges and best practices in industry-academia collaborations in software engineering: a systematic literature review," *Information and Software Technology,* vol. 79, pp. 106–127, 2016.

[27] S. Doğan, A. Betin-Can, and V. Garousi, "Web application testing: a systematic literature review," *Journal of Systems and Software,* vol. 91, pp. 174-201, 2014.

[28] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology,* vol. 64, no. 0, pp. 1-18, 2015.

[29] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, United Kingdom, 2014.

[30] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2008.

[31] V. Garousi, M. Felderer, Ç. M. Karapıçak, and U. Yılmaz, "Testing embedded software: A survey of the literature," *Information and Software Technology, In Press,* 2018.

[32] A. E. K. Sobel, "Software Engineering Education Knowledge (SEEK)," *http://sites.computer.org/ccse/know/FinalDraft.pdf,* 2003, Last accessed: Feb. 2019.

[33] Institute for Electrical and Electronic Engineers (IEEE-CS) and Association for Computing Machinery (ACM), "Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering: Software Engineering SE) 2004," in *Computing Curricula Series, http://sites.computer.org/ccse/,* 2004.

[34] V. Garousi and M. V. Mäntylä, "Citations, research topics and active countries in software engineering: A bibliometrics study " *Elsevier Computer Science Review,* vol. 19, pp. 56-77, 2016.

[35] V. Garousi, "A bibliometric analysis of the Turkish software engineering research community," *Springer Journal on Scientometrics,* vol. 105, no. 1, pp. 23-49, 2015.

[36] N. R. Haddaway, A. M. Collins, D. Coughlin, and S. Kirk, "The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching," *PLoS ONE,* vol. 10, no. 9, 2015.

[37] P. Jacsó, "Google Scholar: the pros and the cons," *Online information review,* vol. 29, no. 2, pp. 208-214, 2005.

[38] J. Zhi, V. Garousi, B. Sun, G. Garousi, S. Shahnewaz, and G. Ruhe, "Cost, benefits and quality of software development documentation: a systematic mapping," *Journal of Systems and Software,* vol. 99, pp. 175–198, 2015.

[39] V. Garousi, Y. Amannejad, and A. Betin-Can, "Software test-code engineering: a systematic mapping," *Journal of Information and Software Technology,* vol. 58, pp. 123–147, 2015.

[40] F. Häser, M. Felderer, and R. Breu, "Software paradigms, assessment types and non-functional requirements in model-based integration testing: a systematic literature review," presented at the Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, 2014.

[41] M. Felderer, P. Zech, R. Breu, M. Büchler, and A. Pretschner, "Model-based security testing: a taxonomy and systematic classification," *Software Testing, Verification and Reliability,* 2015.

[42] S. Mohacsi, M. Felderer, and A. Beer, "Estimating the Cost and Benefit of Model-Based Testing: A Decision Support Procedure for the Application of Model-Based Testing in Industry," in *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*, 2015, pp. 382-389.

[43] K. Godin, J. Stapleton, S. I. Kirkpatrick, R. M. Hanning, and S. T. Leatherdale, "Applying systematic review search methods to the grey literature: a case study examining guidelines for school-based breakfast programs in Canada," *Systematic Reviews,* vol. 4, pp. 138-148, 2015.

[44] Q. Mahood, D. Van Eerd, and E. Irvin, "Searching for grey literature for systematic reviews: challenges and benefits," *Research Synthesis Methods,* vol. 5, no. 3, pp. 221-234, 2014.

[45] J. Adams *et al.*, "Searching and synthesising 'grey literature' and 'grey information' in public health: critical reflections on three case studies," *Systematic Reviews,* journal article vol. 5, no. 1, p. 164, 2016.

[46] V. Garousi and M. V. Mäntylä, "When and what to automate in software testing? A multivocal literature review," *Information and Software Technology,* vol. 76, pp. 92-117, 2016.

[47] V. Garousi, M. Felderer, and T. Hacaloğlu, "Software test maturity assessment and test process improvement: A multivocal literature review," *Information and Software Technology,* vol. 85, pp. 16–42, 2017.

[48] T. Greenhalgh and R. Peacock, "Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources," *Bmj,* vol. 331, no. 7524, pp. 1064-1065, 2005.

[49] C. Wohlin and B. Regnell, "Achieving industrial relevance in software engineering education," in *Proceedings of Conference on IEEE Software Engineering Education and Training*, 1999, pp. 16-25.

[50] S. R. Faulk, "Achieving industrial relevance with academic excellence: lessons from the Oregon Master of Software engineering," in *Proceedings of International Conference on Software Engineering*, 2000, pp. 293-302.

[51] J. H. Benamati, Z. D. Ozdemir, and H. J. Smith, "Aligning undergraduate IS curricula with industry needs," *Communications of the ACM,* vol. 53, no. 3, pp. 152-156, 2010.

[52] Y. Kim, J. Hsu, and M. Stern, "An update on the IS/IT skills gap," *Journal of Information Systems Education,* vol. 17, no. 4, p. 395, 2006.

[53] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer London, 2008, pp. 285-311.

[54] M. B. Miles, A. M. Huberman, and J. Saldana, *Qualitative Data Analysis: A Methods Sourcebook*, Third Edition ed. SAGE Publications Inc., 2014.

[55] J. Miller, "Applying meta-analytical procedures to software engineering experiments," *Journal of Systems and Software,* vol. 54, no. 1, pp. 29-39, 2000.

[56] G. V. Glass, "Primary, secondary, and meta-analysis of research," *Educational researcher,* vol. 5, no. 10, pp. 3-8, 1976.

[57] R. Seiler, "Getting Results With 360 Assessments," *Law Enforcement Technology,* vol. 32, no. 9, pp. 148,150,152-156, 2005.

[58] R. M. Groves, *Survey errors and survey costs*. John Wiley & Sons, 2004.

[59] A. Alain, B. Pierre, D. Robert, and W. M. James, *Guide to the Software Engineering Body of Knowledge (SWEBOK), version 1*. IEEE Press, 2001.

[60] A. Alain, B. Pierre, D. Robert, and W. M. James, *Guide to the Software Engineering Body of Knowledge (SWEBOK), version 2*. IEEE Press, 2004.

[61] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser, "SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," *IEEE Computer,* vol. 48, no. 11, pp. 106-109, 2015.

[62] Association for Computing Machinery (ACM), "Computer Engineering Curricula 2004," in *https://www.acm.org/binaries/content/assets/education/curricula-recommendations/ce-final-report.pdf*, 2004.

[63] Association for Computing Machinery (ACM), "Computer Engineering Curricula 2016," in *https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf*, 2016.

[64] A. f. C. M. Joint Task Force on Computing Curricula and I. C. Society, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, 2013, p. 518.

[65] M. Sabin, H. Alrumaih, J. Impagliazzo, B. M. Lunt, C. Tang, and M. Zhang, "ACM/IEEE-CS Information Technology Curriculum 2017: A Status Update," presented at the Proceedings of the 16th Annual Conference on Information Technology Education, Chicago, Illinois, USA, 2015.

[66] H. Topi *et al.*, "Curriculum Guidelines for Undergraduate Degree Programs in Information Systems," *ACM,* 2010.

[67] Accreditation Board for Engineering and Technology (ABET), "ABET," in *http://www.abet.org/accreditation/*, Last accessed: Feb. 2019.

[68] L. Briand, "Embracing the engineering side of software engineering," *IEEE software,* vol. 29, no. 4, pp. 96-96, 2012.

[69] P. L. Li, A. J. Ko, and J. Zhu, "What makes a great software engineer?," in *Proceedings of International Conference on Software Engineering*, 2015, pp. 700-710.

[70] C. Bereiter and M. Scardamalia, "Education for the Knowledge Age: Design-Centered Models of Teaching and Instruction," in *Handbook of educational psychology*.Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 2006, pp. 695-713.

[71] R. L. Katz, *Skills of an effective administrator*. Harvard Business Review Press, 2009.

[72] B. Boehm, "A view of 20th and 21st century software engineering," in *Proceedings of the international conference on Software engineering*, 2006, pp. 12-29.

[73] D. L. Parnas, "Software engineering programs are not computer science programs," *IEEE software,* vol. 16, no. 6, pp. 19-30, 1999.