

# Closing the Gap Between Software Engineering Education and Industrial Needs

Vahid Garousi, Queen's University Belfast

Görkem Giray, Independent Researcher

Eray Tüzün, Bilkent University

Cagatay Catal, Wageningen University & Research and Bahcesehir University

Michael Felderer, Blekinge Institute of Technology and University of Innsbruck

*// Many recent software engineering graduates often face difficulties when beginning their professional careers, due to misalignment of the skills learned in their university education with what is needed in industry. In this article, we report a literature review of the studies that have been done to make improvements on this issue. //*

**MANY SOFTWARE ENGINEERING** (SE) university programs have evolved from computer science programs and still focus on theoretical and technical computer science topics as well as mathematical foundations. This emphasis seems to cause a discrepancy between the skills learned from an SE university education and those needed in SE employment.<sup>1,2</sup> In the community, some 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.”<sup>3</sup>

We, the authors, are active SE educators who each have been teaching various SE courses for more than 15 years. We also have had active industry experience or have worked in close collaboration with practitioners in joint industry–academia projects. In response to feedback from industry partners who have hired our students and from recent graduates, and the needs of our university departments and SE programs, we decided to conduct a systematic literature review (SLR) to highlight the findings of various studies that discuss aligning SE education with industry needs.

We used the established process for performing SLR studies in SE<sup>4</sup> and systematically gathered a set of 33 papers on this subject, published between 1995 and 2018. Our review strives to identify the most important skills in industry and reveal knowledge deficiencies in graduating SE students.

Illuminating important knowledge gaps for various SE topics ultimately helps to understand how we can best train future software



engineers. By summarizing what we as a community know in this area, this article aims to benefit the readers (both educators and hiring managers) by providing the overall state of the community with respect to aligning SE education with industrial needs and documenting the body of knowledge in this area.

### The Review Procedure

In our review and mapping, we followed the established process for performing SLR studies in SE<sup>4</sup> and used our experience from conducting SLRs in the past.<sup>5</sup> All of the authors conducted each of the steps as a team. We searched the Google Scholar database. Our search string was as follows: (educational needs OR knowledge needs OR desired skills OR essential competencies OR knowledge requirements OR skill requirements) AND (software engineers OR software developers). We address the following review questions:

- What skills are most important in the software industry? Given the rapidly changing nature of SE, we wanted to know if the most important have changed in the last five years.
- Is there evidence of knowledge deficiencies in graduating SE students? What are the topics with highest knowledge deficiencies?
- To what extent are soft skills important, in addition to hard (technical) skills?

We only included papers that focused on aligning SE education with industrial needs and were based on empirical data, such as survey results or interview data. We included the latter criteria to exclude papers based purely on personal opinions. After compiling an initial pool of 94 papers, we

voted systematically using the aforementioned criteria. Our final pool included 33 papers. We used meta-analysis,<sup>6</sup> a research method that is a form of synthesis, which combines the quantitative data from primary studies (the pool of 33 papers in this article) to aggregate the results of primary studies to provide a consolidated overview on a given topic.

We provide a more detailed description of our SLR process and discuss how we identified and addressed the potential threats to validity to our review in an online web extras section<sup>7</sup> that shows the 33 papers in our final pool. All of the data that we have extracted from the papers can be found in an online repository formulated as a Google spreadsheet.<sup>8</sup> In this article, we use the “[Pi]” format to refer to the papers in the pool. The data shows that attention for this topic has risen in recent years (Table 1).

### More Than 4,000 Data Points From 12 Countries

Most of the papers in the pool had extracted data from one country only, e.g., [P2] had data from the United Kingdom and [P8 ... S11] had data from the United States. The advantage of our metasynthesis (meta-analysis) is that the combined data set has data from 12 countries, which provides stronger evidence on the subject than the single-country studies. The top countries from which data were gathered were the United States (15 papers), Canada (four papers), South Africa (four papers), New Zealand (two papers), and Spain (two papers). The United Kingdom, Norway, Philippines, Jordan, Australia, Finland, and Samoa were each represented in one paper. Two papers had data from both the United States and Canada, and one paper surveyed worldwide data.

The number of data points (survey respondents) varied, with studies that had between eight [P28] and 600 respondents [P21]. Since the studies were primarily conducted in different countries, there is slim chance that a single software engineer could have participated in more than one study in the pool. Thus, when we add up the number of respondents from all 33 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 including such a large combined data set, our study aims to provide a comprehensive overview.

### The Most Important Skills in the Industry

The questionnaires designed for and used by the studies had differences with respect to the concrete SE topics used in them. In other words, when asking respondents to rate (rank) the importance of SE topics, different papers used different sets of SE topics. Six studies used the SE topics as proposed in different versions of the SE Body of Knowledge (SWEBOK)<sup>9</sup> (version 1.0 developed in 1999, version 2.0 in 2004, and version 3.0 in 2014) [P1, S3, S4, S6, S14, S32]. Two studies [P3, S26] used a similar guideline from the IEEE, called the SE Education Knowledge, which was developed in 2004. [P4] used the Association for Computing Machinery (ACM) SE 2004 curriculum guideline. [P26] used the ACM Body of Knowledge of Computing Curriculum for Computer Science. Three other studies used the ACM IT curriculum and three used the ACM Information Systems (IS) curriculum. The remaining 20 studies did not use a single curriculum model, but instead synthesized the list of SE topics either from the literature or by an initial interview with practitioners.

Table 1. A list of the studies reviewed in this meta-analysis.

ID	Paper reference
[P1]	T. C. Lethbridge, "A survey of the relevance of computer science and software engineering education," in <i>Proc. Conf. Software Engineering Education</i> , 1998, pp. 56–66.
[P2]	I. C. Mow, H. Sasa, F. Maua-Faamau, E. Mauai, and M. Tanielu, "An evaluation of relevance of computing curricula to industry needs," <i>Systemics, Cybernetics, Informatics</i> , vol. 13, no. 1, pp. 7–12, 2015.
[P3]	B. Kitchenham, D. Budgen, P. Brereton, and P. Woodall, "An investigation of software engineering curricula," <i>J. Syst. Softw.</i> , vol. 74, no. 3, pp. 325–335, 2005.
[P4]	A. Deak and G. Sindre, "Analyzing the importance of teaching about testing from alumni survey data," in <i>Proc. Norwegian Informatics Conf.</i> , 2013. [Online]. Available: <a href="http://www.nik.no/2013/3-1-Deak_Sindre_NIK_2013.pdf">http://www.nik.no/2013/3-1-Deak_Sindre_NIK_2013.pdf</a>
[P5]	C. Watson and K. Blincoe, "Attitudes towards software engineering education in the New Zealand industry," in <i>Proc. Annu. Conf. Australasian Association for Engineering Education</i> , 2017, pp. 785–792.
[P6]	R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F. J. García-Peñalvo, and E. Tovar-Caro, "Competence gaps in software personnel: A multi-organizational study," <i>Comp. Human Behav.</i> , vol. 29, no. 2, pp. 456–461, 2013.
[P7]	M. E. McMurtrey, J. P. Downey, S. M. Zeltmann, and W. H. Friedman, "Critical skill sets of entry-level IT professionals: An empirical examination of perceptions from field personnel," <i>J. Inform. Technol. Ed.: Res.</i> , vol. 7, no. 1, pp. 101–120, Jan. 2008.
[P8]	D. M. Lee, E. M. Trauth, and D. Farwell, "Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation," <i>MIS Quart.</i> , vol. 19, no. 3, pp. 313–340, 1995.
[P9]	K. Jones, L. N. Leonard, and G. Lang, "Desired skills for entry level IS positions: Identification and assessment," <i>J. Comp. Inf. Syst.</i> , vol. 58, no. 3, pp. 214–220, 2018.
[P10]	R. D. Howard, "Does the information systems curriculum meet business needs: Case study of a southeastern college," Ph.D. dissertation, School Business Technol., Capella Univ., Minneapolis, MN, 2017.
[P11]	C. Scaffidi, "Employers' need for computer science, information technology and software engineering skills among new graduates," <i>Int. J. Comp. Sci., Eng. Inform. Technol.</i> , vol. 8, no. 1, pp. 1–12, 2018.
[P12]	F. Patacsil and C. L. S. Tablatin, "Exploring the importance of soft and hard skills as perceived by IT internship students and industry: A gap analysis," <i>J. Technol. Sci. Ed.</i> , 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," Ph.D. dissertation, Dept. Computer Sci., North Dakota State University, Fargo, 2012.
[P14]	R. Colomo-Palacios, E. Tovar-Caro, Á. García-Crespo, and J. M. Gómez-Berbis, "Identifying technical competences of IT professionals: the case of software engineers," <i>Int. J. Human Capital Inform. Technol. Prof.</i> , vol. 1, no. 1, pp. 31–43, 2010.
[P15]	G. Lang, K. Jones, and L. N. Leonard, "In the know: Desired skills for entry-level systems analyst positions," <i>Iss. Inform. Syst.</i> , vol. 16, no. 1, pp. 142–148, 2015.
[P16]	M. Stevens and R. Norman, "Industry expectations of soft skills in IT graduates: A regional survey," in <i>Proc. Australasian Comp. Sci. Conf.</i> , 2016, pp. 13–21.
[P17]	J. Liebenberg, M. Huisman, and E. Mentz, "Industry's perception of the relevance of software development education," <i>J. Transdisciplinary Res. Southern Africa</i> , vol. 11, no. 3, pp. 260–284, 2015.
[P18]	A. Radermacher, G. Walia, and D. Knudson, "Investigating the skill gap between graduating students and industry expectations," in <i>Companion Proc. Int. Conf. Software Engineering</i> , 2014, pp. 291–300.
[P19]	C. L. Aasheim, L. Li, J. D. Shropshire, and C. A. Kadlec, "IT program curriculum recommendations based on a survey of knowledge and skill requirements for entry-level IT workers," in <i>Proc. Southeastern INFORMS Conf.</i> , 2011, pp. 209–219.

(Continued)

**Table 1. A list of the studies reviewed in this meta-analysis (cont.).**

ID	Paper reference
[P20]	C. V. Bullen, T. Abraham, K. Gallagher, J. C. Simon, and P. Zwieg, "IT workforce trends: Implications for curriculum and hiring," <i>Commun. Assoc. Inform. Syst.</i> , vol. 20, pp. 545–554, 2007.
[P21]	C. L. Aasheim and 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?" <i>J. Inform. Syst. Educ.</i> , vol. 20, no. 3, pp. 349–356, 2009.
[P22]	C. L. Aasheim, S. Williams, and E. S. Butler, "Knowledge and skill requirements for IT graduates," <i>J. Comp. Inform. Syst.</i> , vol. 49, no. 3, pp. 48–53, 2009.
[P23]	J. Liebenberg, M. Huisman, and E. Mentz, "Knowledge and skills requirements for software developer students," <i>Int. J. Social, Behav., Ed., Econ., Bus. Ind. Eng.</i> , 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," <i>CrossTalk: J. Def. Softw. Eng.</i> , vol. 28, no. 1, pp. 4–8, 2015.
[P25]	T. C. Lethbridge, "Priorities for the education and training of software engineers," <i>J. Syst. Softw.</i> , vol. 53, no. 1, pp. 53–71, 2000.
[P26]	S. Hanna, H. Jaber, A. Almasalmeh, and F. A. Jaber, "Reducing the gap between software engineering curricula and software industry in Jordan," <i>J. Softw. Eng. Applic.</i> , vol. 7, pp. 602–616, June 2014.
[P27]	O. Minor and J. Armarego, "Requirements engineering: A close look at industry needs and a model curricula," <i>Australasian J. Inform. Syst.</i> , vol. 13, no. 1, pp. 192–208, 2005.
[P28]	J. Liebenberg, M. Huisman, and E. Mentz, "Software: University courses versus workplace practice," <i>Ind. Higher Ed.</i> , vol. 29, no. 3, pp. 221–235, 2015.
[P29]	A. Begel and B. Simon, "Struggles of new college graduates in their first software development job," <i>Assoc. Comput. Machinery Special Interest Group Comput. Sci. Edu. Bull.</i> , vol. 40, no. 1, pp. 226–230, 2008.
[P30]	J. Liebenberg, M. Huisman, and E. Mentz, "The relevance of software development education for students," <i>IEEE Trans. Edu.</i> , vol. 58, no. 4, pp. 242–248, 2015.
[P31]	T. C. Lethbridge, "The relevance of software education: A survey and some recommendations," <i>Ann. Softw. Eng.</i> , vol. 6, no. 1–4, pp. 91–110, 1998.
[P32]	A. Seffah, "Training developers in critical skills," <i>IEEE Softw.</i> , vol. 16, no. 3, pp. 66–70, 1999.
[P33]	T. C. Lethbridge, "What knowledge is important to a software professional?" <i>Computer</i> , vol. 33, no. 5, pp. 44–50, 2000.

With such diverse SE topics used in the studies, we selected the most relevant model, SWEBOK version 3.0. We mapped the SE topics discussed in the papers to the 15 SWEBOK knowledge areas (KAs), which are as follows:

- 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
- mathematical foundations.

The next step was consolidating the quantitative data of skill (topic) importance from all of the papers; almost all of them had presented ranking of the most important skills. To be able to cross-compare and synthesize data in a

consolidated way, we harmonized the importance ranking data as follows. We normalized the topic rankings in each paper to the range of [0, 1] for each SWEBOK KA. For example, for [P1], three of the 14 ranked topics related to the design KA, including general architecture (ranked 1), object-oriented design (ranked 9), and user-interface design (ranked 12). We calculated the average of (1, 9, 12), which equals 7.33, and divided it by 14, the number of all SE topics in that paper. The normalized rank metric was

0.52. Since rank data were used, the lower the value of this metric, the higher the importance of a given topic. Thus, by calculating normalized rank, we aggregated the normalized importance (0.48 in the previous example). Once we had the importance metric of each KA for each paper, we calculated its average among all papers.

Figure 1 shows the normalized importance metrics of each topic and the number of papers that it has appeared in as a scatter plot. Given the fast-changing nature of the SE field and its KAs, we were eager to compare the skill importance data from all of the papers against those published in the last five years, so we calculated the aforementioned metrics for each case separately.

Comparison of the two charts in Figure 1 provides interesting insights. When reviewing all of the papers, the requirements, design, and testing are most important and are frequently mentioned in SE professional practice, with project management and development listed next. However, when considering recent papers, the top-three topics become SE professional practice, project management, and testing. This ranking seems to mean that less-technical skills, such as SE professional practice and project management, have become even more important in recent years and cover topics such as professionalism, group dynamics, and communication skills. These soft skills are especially required in modern agile software development, which is more strongly based on communication and interaction than traditional waterfall approaches. In our experience, an effective approach for covering project management and SE professional practice in education is with larger SE projects

done by student teams, either in class or even together with companies.<sup>10</sup>

Mathematical and engineering foundations, as well as SE economics, rank 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. Adopting ideas from these subjects offers new approaches to solving problems in engineering software.<sup>11</sup> In line with this finding, it is interesting to observe that requirements, testing, and design are considered more important than actual development. However, in our experience, this is not always reflected in SE education, especially if it is embedded into computer science curricula.

### Knowledge Gaps: Highlighting the Topics That We Should Teach More

In quantitative terms, eight of the 33 studies also measured the knowledge gap (deficiency) from their survey participant responses, 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 his or her university education. We extracted the quantitative knowledge-gap values and calculated their normalized average. In Figure 2, we show a scatter plot to visualize the average knowledge-gap values versus their importance. The  $x$  axis shows the average importance and the  $y$  axis shows the average knowledge gap. In all eight papers, the two factors were shown to be quite correlated and, with increasing reported importance, more of a knowledge gap has generally been reported. The greatest reported knowledge gaps are

in the areas of configuration management, SE models and methods, SE process, design (and architecture), and testing. Thus, in general, university programs and companies that are training newly hired staff will focus on these topics. We have also divided the scatter plot of Figure 2 into four quadrants to clearly see the SE topics with low or high importance and a low or high knowledge gap.

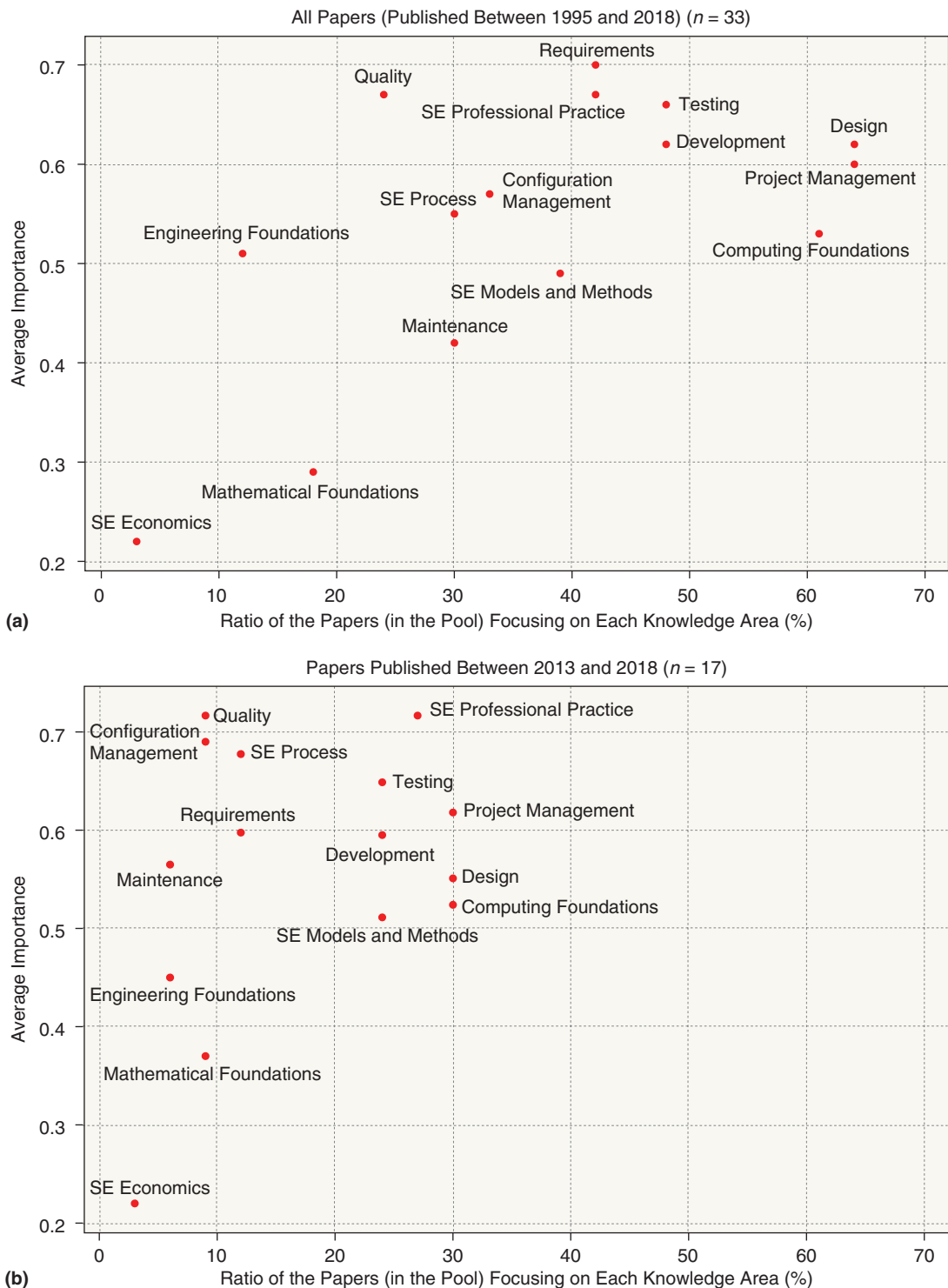
Topics in Q1 (high importance, high gap) are those that require the most attention with respect to the need for improvements in SE education in university programs. They have high importance but also have a high knowledge gap. Topics in Q2 (low importance, high gap) should be the focus next with respect to SE education (after those in Q1). They have relatively low importance, but high knowledge gaps in those topics remain and thus need attention for more education and training on those topics.

For topics in Q3 (high importance, low gap), university programs are generally doing a good job, since knowledge gaps in these topics are relatively low, while they are quite important with respect to technical needs in the industry. Only the software development topic falls slightly in Q3 of one of the scatter plots.

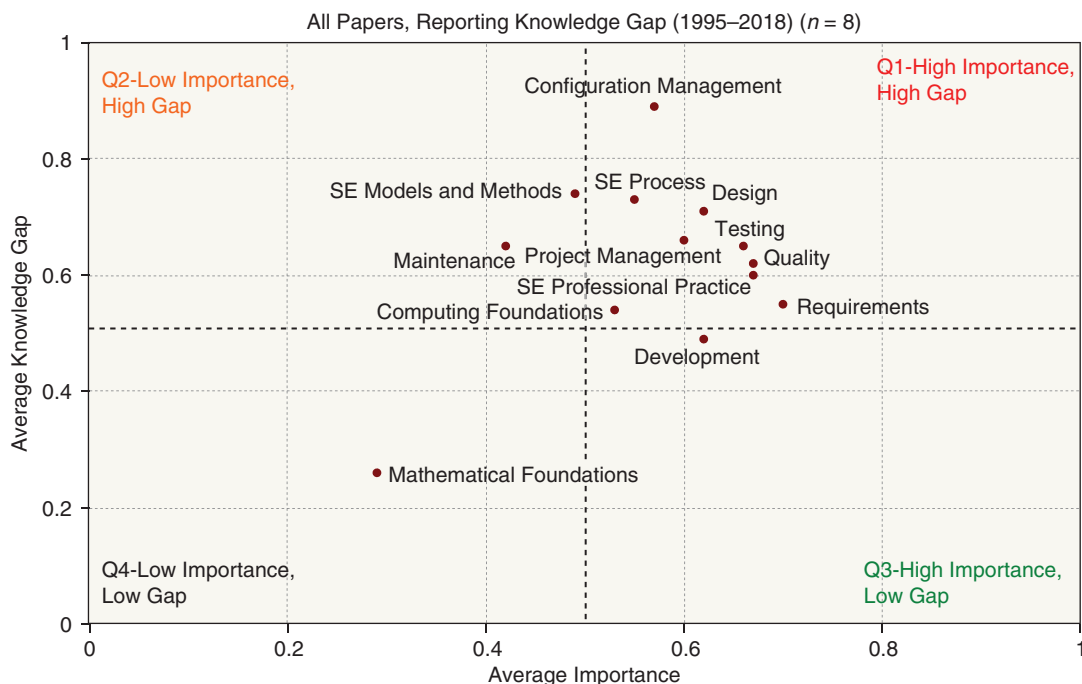
Topics in Q4 have low importance and a low knowledge gap, so they are least in need of improvements and the attention of SE education in university programs. The mathematical foundations KA falls into Q4 in both scatter plots.

### Hard Skills Alone Are Not Enough: Do You Have Soft Skills?

It is widely discussed in the community that hard (technical) skills alone do not make a great software



**FIGURE 1.** The most important skills: (a) data from all the papers versus (b) papers published in the last five years only.



**FIGURE 2.** The topics with the greatest knowledge gap—where importance (usage) of an SE topic exceeds current knowledge of survey participants.

engineer<sup>12</sup> and that soft skills are equally important (if not more). Hard skills are composed of domain knowledge and technical skills, while soft skills are composed of team and interpersonal skills. “Soft skills contribute significantly to individual learning, team performance, client relations and awareness of the business context” [P16].

In 24 of the 33 studies, the importance of soft skills was recognized. We categorized soft skills as teamwork and communication (discussed in 19 studies), leadership (13 studies), critical thinking (11 studies), and others (17 studies). Other important soft skills such as cultural fit, understanding of business drives, aptitude, attitude, coping with ambiguity, learning and curiosity, and passion/drive to innovate were also mentioned.

One of the studies, [P16], specifically focused on industry expectations of soft skills in IT graduates. The data came from a regional survey conducted in New Zealand in 2016. Key findings from the study that are of interest to educators are as follows. While in-house technical training is widely used to advance graduate skills and teach new technologies, most employers consider these soft skills to be untrainable in the workplace, making them the critical hurdle for employment. Furthermore, studies show that short-term pressure on employers for technical skills can result in overlooking soft skills. One interesting quote from the study is, “The public sector especially needs engineers with a sophisticated understanding of the social environment within which their

activity takes place, a systems understanding, and an ability to communicate with stakeholders.” Another is, “Today’s working environment is all about relationships, both internal and external. We need people who can step-up and be accountable without always needing a coach/mentor standing by. People working in isolation contribute more errors than teams.”

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 makeup seven of the top eight most important skills [in that study].” While, “soft



**VAHID GAROUSI** is a senior lecturer (associate professor) of software engineering at Queen's University Belfast, United Kingdom. In parallel to his academic career, he is a software engineering consultant and coach and provides consultancy and corporate training services. His research interests include software engineering, software testing, empirical studies, action research, and industry-academia collaborations. He is passionate about successfully bridging industry and academia in both research and education of software engineering. Garousi received a Ph.D. in software engineering from Carleton University, Ottawa, Canada, in 2006. He was selected as a Distinguished Speaker for the IEEE Computer Society from 2012 to 2015. Contact him at [v.garousi@qub.ac.uk](mailto:v.garousi@qub.ac.uk).



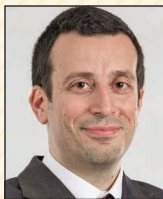
**CAGATAY CATAL** is a faculty member at Bahcesehir University and guest professor at Wageningen University & Research. His research interests include software engineering, education research, and data mining. Catal received a Ph.D. in computer engineering from Yildiz Technical University. Contact him at [cagatay.catal@wur.nl](mailto:cagatay.catal@wur.nl).



**GÖRKEM GIRAY** is a software engineer and an independent researcher. His research interests include software engineering and education research. Giray received a Ph.D. in computer engineering from Ege University. Contact him at [gorkemgiray@gmail.com](mailto:gorkemgiray@gmail.com).



**MICHAEL FELDERER** is an associate professor at the University of Innsbruck and a guest professor at the Blekinge Institute of Technology. His research interests include software quality, processes, analytics, empirical methods, and education research in software engineering. Felderer received a Ph.D. in computer science from the University of Innsbruck. Contact him at [michael.felderer@uibk.ac.at](mailto:michael.felderer@uibk.ac.at).



**ERAY TÜZÜN** is a faculty member at Bilkent University. His research interests include software analytics, software reuse, empirical software engineering, and software engineering education. Tüzün received a Ph.D. in information systems from Middle East Technical University. Contact him at [eraytuzun@cs.bilkent.edu.tr](mailto:eraytuzun@cs.bilkent.edu.tr).

skills and business skills must be included in curricula,” study [P22] recommends.

These statements are in agreement with our finding that the knowledge area of SE professional practice is of high importance (see Figure 1), which is comprised of topics such as professionalism, group dynamics, and communication skills.

### Other Interesting Findings

We observed many other interesting findings when reviewing the papers. For example, there were suggestions for decreasing an emphasis on certain topics in SE university education (i.e., what we should teach less). [P1] expressed that as, “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 “overemphasis on mathematical topics and underemphasis on business topics” in SE education. [P3] called for less of an educational focus on parsing and compiler design, formal specification methods, digital electronics, and digital logic in SE programs.

Going further, some studies discussed how determining the amount of coverage each SE topic should have is not enough and that educators should teach using “real-world” software system examples. For example, [P27] reported that, “Real-life and practical experience must be included in students’ education.” [P26] also highlighted the need for “more exposure to real life, exercises, team assignments or industry projects.” Some of the authors have had experience in such ideas.<sup>10</sup>

Other interesting suggestions were made in [P28], as follows. “Instead of a greenfield project, a more valuable experience would provide students a large preexisting codebase to which they must fix bugs (injected or real) and write additional features. Also valuable would be a management component, in which 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, challenge them to solve bugs several times until the “right” fix is found, or who give them sometimes capricious and cryptic weekly commandments on requirements or testing that they must puzzle out and solve together as a team.” The authors of this article often heard similar comments when talking to experienced SE practitioners.


### Implications and the Road Ahead

The findings presented in this article show the importance of an SE

professional practice and soft skills in general. These include the importance of certain SE activities and skills in SE education (especially requirements for engineering, design, and testing), knowledge gaps in specific areas of SE (especially configuration management, SE models and methods, and 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. 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 that have the greatest knowledge gaps. Also, in the context of a large software company in Turkey with which one of the authors was affiliated, an industrial training program for potential new hires was recently conducted<sup>13</sup> based on the insights provided by this review study. We are certain that the results and findings presented in this article 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 information 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 separate computer science from SE university programs.<sup>14</sup> 

### References

1. A. D. Radermacher, “Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectations of industry,” M.S. thesis, Dept. Computer Sci., North Dakota State Univ., Fargo, ND, 2012.
2. J. de Rojas, “Mind the gap: A report on the UK’s technology skills landscape,” Hired. Accessed on: Mar. 21, 2019. [Online]. Available: [hired.com/skills-gap](https://hired.com/skills-gap)
3. J. Baker, “2018’s software engineering talent shortage—It’s quality, not just quantity,” Hackernoon, 2017. [Online]. Available: [goo.gl/MVwcqX](https://goo.gl/MVwcqX)
4. B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” School of Computer Science and Mathematics, Keele University, U.K., EBSE (Evidence-Based Software Engineering) Tech. Rep. EBSE-2007-01, July 9, 2017. [Online]. Available: [https://www.elsevier.com/\\_data/promis\\_misc/525444systematicreviewsguide.pdf](https://www.elsevier.com/_data/promis_misc/525444systematicreviewsguide.pdf)
5. V. Garousi, M. Felderer, and T. Hacaloğlu, “What we know about software test maturity and test process improvement,” *IEEE Softw.*, vol. 35, no. 1, pp. 84–92, 2018.
6. D. S. Cruzes and T. Dyba, “Synthesizing evidence in software engineering research,” in *Proc. Int. Symp. Empirical Software Engineering and Measurement*, 2010, pp. 1–10.
7. V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, “The web extras section of the article ‘closing the gap between software engineering education and industrial needs’.”

Accessed on: Mar. 21, 2019. [Online]. Available: [goo.gl/TBEQwu](http://goo.gl/TBEQwu)

8. V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, "SLR—Aligning SE education with industry needs." Accessed on: Mar. 21, 2019. [Online]. Available: [goo.gl/kpx6c7](http://goo.gl/kpx6c7)
9. P. Bourque and R. E. Fairley, *Guide to the Software Engineering Body of Knowledge*. Washington, DC: IEEE Computer Society Press, 2014.
10. V. Garousi, "Incorporating real-world industrial testing projects in software testing courses: Opportunities, challenges, and lessons learned," in *Proc. IEEE Conf. Software Engineering Education and Training*, 2011, pp. 396–400.
11. L. Briand, "Embracing the engineering side of software engineering," *IEEE Softw.*, vol. 29, no. 4, pp. 96–96, 2012.
12. P. L. Li, A. J. Ko, and J. Zhu, "What makes a great software engineer?" in *Proc. Int. Conf. Software Engineering*, 2015, pp. 700–710.
13. 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 *Proc. Int. Conf. Software Engineering:*

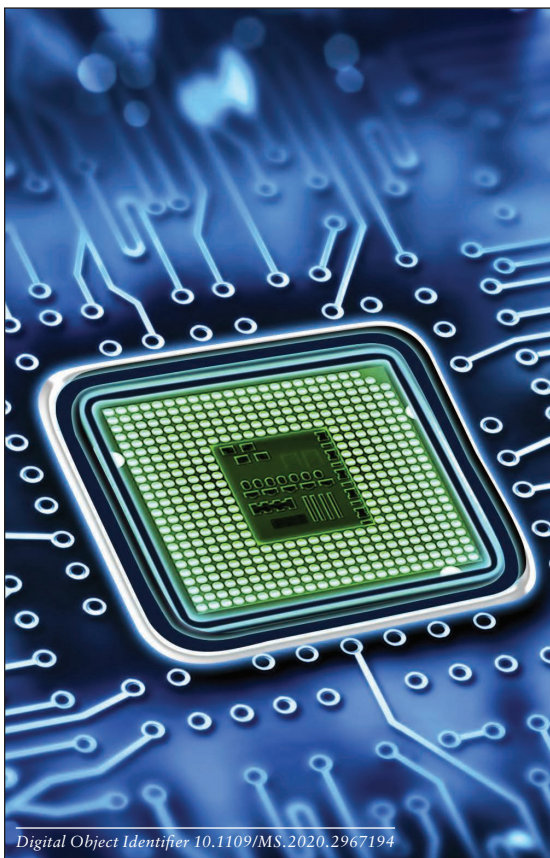
*Software Engineering Education and Training*, 2018, pp. 68–77.

14. D. L. Parnas, "Software engineering programs are not computer science programs," *IEEE Softw.*, vol. 16, no. 6, pp. 19–30, 1999.



IEEE COMPUTER SOCIETY  
**DIGITAL LIBRARY**

Access all your IEEE Computer Society subscriptions at  
**computer.org**  
**/mysubscriptions**



Digital Object Identifier 10.1109/MS.2020.2967194

IEEE TRANSACTIONS ON

# COMPUTERS

## Call for Papers: *IEEE Transactions on Computers*

Publish your work in the IEEE Computer Society's flagship journal, *IEEE Transactions on Computers*. The journal seeks papers on everything from computer architecture and software systems to machine learning and quantum computing.

Learn about calls for papers and submission details at  
[www.computer.org/tc](http://www.computer.org/tc).

