Turk. J. Math. Comput. Sci.
12(2)(2020) 92–100

DOI : 10.47000/tjmcs.789945

The Analysis of Mathematical Skills Used in the Software Industry

Deniz Akdur

ASELSAN Inc., Ankara, Turkey.

Received: 03-09-2020 • Accepted: 24-09-2020

Abstract. Mathematical skills can provide a scientific basis for software engineering (SE) activities by helping practitioners comprehend the unambiguous logic, which is critical for both abstract thinking and problem solving besides development of any applications in the software industry. Many practitioners in different roles take different mathematics courses at the university education; however there are various perceptions about the necessity and the weight of these courses. *"What are their usage in the software industry?*" or *"do we need such a (heavy) mathematical curriculum at the university?*" are frequently asked questions for software teams. To better understand different opinions, we utilized the results of our survey, which investigated the skills gaps in the software industry. 628 Turkey-educated practitioners, whose undergraduate degree was completed in Turkey, working in 13 countries responded the survey. In this paper, we examined different perceptions (1) about the usage of the mathematics courses in the software industry; and (2) about current weight of these courses in the curriculum. The findings shed light on the perceptions about mathematical skills in the industry by presenting various cross-factor analysis.

2010 AMS Classification: 97NXX, 68TXX, 68NXX

Keywords: Curriculum, Mathematics courses, Software engineering education, Software industry, Survey.

1. Introduction

Software engineering (SE) incorporates both organizational and technical aspects, where a software professional works with various description and implementation techniques with modeling. Mathematical techniques can provide a scientific basis for software practitioners, who can program only for something that follows an unambiguous logic. According to Software Engineering Body of Knowledge (SWEBOK), this can be achieved through The Mathematical Foundations knowledge area (KA), which helps software professionals comprehend this logic [6].

Many practitioners in different software engineering roles take different mathematics courses at the university education since SE requires mathematical fundamentals. Concerning the academic curriculum of SE-related degrees (e.g., for both computing and non-computing disciplines), the weight of mathematics courses and their importance might be vary. However, *"how much mathematics is important for software engineers?"*, *"what are the usage of mathematics courses taken at the university education?"* or *"do software engineers need such a (heavy) mathematical curriculum at the university?"* are frequently asked question for the software teams in the industry. In the software industry, there are various perceptions about the necessity and emphasis of mathematics courses: (1) At one extreme, some software practitioners claim that they never 'use' anything related to the mathematics they learned during the university education. According to [7], those practitioners might be right since they don't directly use the mathematics courses (e.g., the rule of integration in calculus); on the other hand, they might be also wrong since the main benefit of the mathematics at the university was the experience of analytical thinking with reasoning and abstraction; (2) Some other

Email address: denizakdur@aselsan.com.tr (D. Akdur)

software engineers think that they do not need mathematics in most cases but knowing mathematics helps build important problem-solving skills. According to them, without knowing the specifics of all the underlying mathematics and just knowing how to apply the concepts (e.g., by using an already-built-in library for mathematical foundations) is enough to be productive and useful. Moreover, such software practitioners claimed that (sufficient) mathematical skills is helpful while modeling in the embedded industry [2]; (3) At the other extreme, some software professionals think that mathematical skills can provide critical insights into problems and increase value of the programs depending on the characteristics of the software developed. For instance, depending on the application (e.g., web, algorithmic works or data science), the focus and the relation of software with mathematics might be vary (i.e., design issues for web is less directly tied to mathematics; however, solving a complex problem with a graph theory or a search algorithm is directly tied to mathematical skills).

It is important to analyze the opinions of the practitioners about the usage of mathematics to make necessary curriculum adjustments depending on industrial needs. To better construct an empirical evidence among different perceptions about the necessity of mathematics (either direct or indirect), we utilized the results of a survey, which we designed to explore various skills gaps in the software industry [3]. 628 Turkey-educated software practitioners, who completed their undergraduate degree (i.e., BSc) in Turkey, working in 13 countries responded the survey. The overall results of this survey will benefit both the software industry and as well as the academia, by creating awareness of the expected skillset and the corresponding gaps in the academic curriculum (mostly for Turkish institutions). In this study, we shed light on the usage of mathematics courses taken at the university education with their coverage in the academic curriculum by presenting various cross-factor analysis. Moreover, we also present the various perceptions of software practitioners about the current weight of mathematics courses in the academic curriculum. We believe that such a study help academia to better adjust the corresponding curriculum according to the opinions of the practitioners.

The remainder of this paper is structured as follows. Section 2 discusses the brief overview of the survey by also presenting the related work for mathematics skills in the industry. Section 3 gives the related work. Section 4 presents the usage and the coverage of mathematics courses besides giving the opinions about the weight of mathematics during university education with various cross-factor analysis. Section 5 discusses on our findings. Section 6 reviews the potential validity threats. Finally, Section 7 concludes this study.

2. Brief Overview of The Survey

Examining the skills gap is crucial for both practitioners and also academics. For employers, hiring properly trained practitioners allows them to spend less time into the workforce; on the other hand, knowing the necessary skillset is critical to make curriculum changes and improving mutual understanding increases industry-academia collaborations (IACs). To achieve these objectives, we designed and conducted a practitioner survey. The main goal was to close the gap between academia and industry. Note that, in this study, we present a subset of 'technical skills' analysis by focusing on mathematical skills.

As a research methodology, we chose the online survey method [10]. While designing survey questions, we conducted expert opinion interviews to select items and arguments in the survey (e.g., the selection of mathematics courses) [3]. Before distributing the survey, we performed pilot studies to prevent misinterpretations in large-scale data collection [11]. According to feedback, we modified some wordings to get common understanding or combine some items (e.g., instead of using only 'logic', we changed this course name by including '(Propositional/Predicate) Logic'. We executed the survey in spring 2019. List of the questions can be found in [5].

The survey was responded by 628 software practitioners, whose undergraduate degrees were completed in Turkey, working in different 13 countries with different demographics data. The results showed that the most important skills are directly related to various factors such as profiles of the practitioners (e.g., SE role(s), work experience) and profiles of the product developed by the practitioner (e.g., application type(s) and industrial sector(s)).

3. Related Work

In the literature, there are few studies, which have investigated the mathematical skills in the software industry.

Lethbridge tried to investigate the necessary SE-related topics in Canada and USA [9]. The survey revealed that there was a need for less emphasis on continuous mathematics such as differential equations and calculus.

Kitchenham et al. investigated whether UK universities matched the needs of the industry [8]. Their results confirmed several observations in [9] concerning the over-emphasis of mathematical topics. They suggested that there should be less focus on mathematics [8].

Surakka confirmed the results of the surveys in [8] and [9] on mathematics courses via a survey conducted in Finland [12]. The author suggested that academia should either place less emphasis on mathematics or they should teach mathematics courses, which are more relevant to the industrial expectations [12].

As one of the motivations of this study, there is not any recent study, which presents the gap between mathematical foundations and the software industry; hence, to close the gap in the existing literature (including Turkey data), our results will be helpful for both the industry and (mostly) for Turkish universities.

4. Survey Results

In this section, opinions of 628 software practitioners with different demographics are presented.

4.1. **Demographic of the participants and their companies.** According to the results, the majority of responses (355 responses) are from Turkey, followed by USA (94 responses), The Netherlands (53 responses), Germany (42 responses) as depicted in Figure 1.

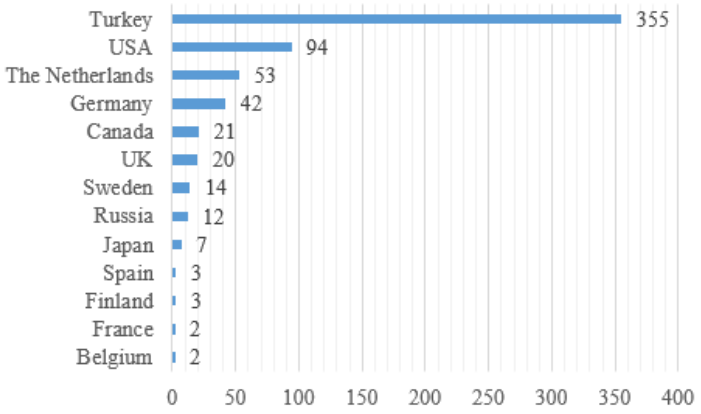


Figure 1. The countries where the participant is working in

The result showed that 58.3% of respondents have a postgraduate degree (either MSc or PhD); whereas and 41.7% of respondents have a BSc degree.

We then asked the university degrees of the participants. Figure 2 shows BSc degrees, in which Electrical and Electronics Engineering (EEE) graduates (one of the non-computing disciplines) have a significant percentage (24.7%). When postgraduate degree(s) are analyzed, the largest group (23.6%) took a CENG curriculum, followed by CS (13.4%), EEE (6.9%) and ECE (5.9%) curriculum.

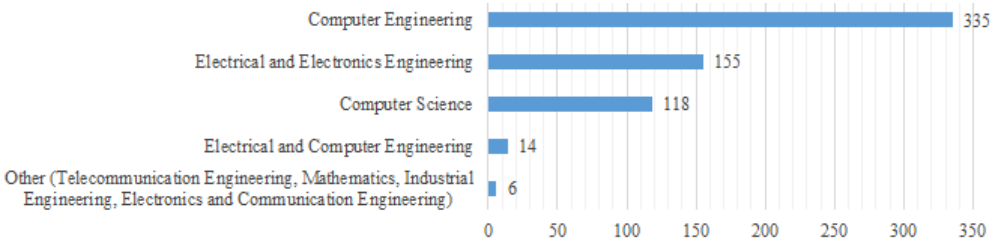
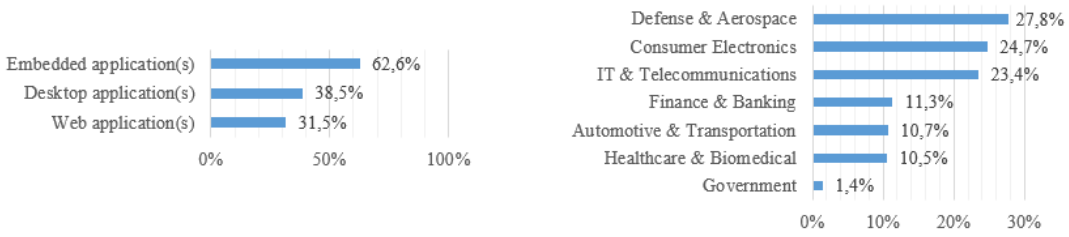


Figure 2. BSc degrees

90.3% of the participants had software developer role; followed by software architect (43.8%). Software project manager (8%), systems engineer (7%), middle/high level management (4.9%), software tester (3.7%) and quality assurance engineer roles were the other responses (Note that this was a multiple-response question; a participant can choose multiple roles in her/his career).

Embedded applications is the majority (62.6%) application type developed by the participant, which is followed by desktop and web (Figure 3.a). The target sectors of the products, which had again multiple-response answers, are depicted in Figure 3.b. 'Defense & Aerospace', 'Consumer Electronics', and 'IT & Telecommunications' are the top three industrial sectors.



(a) Application types (b) Industrial sector

FIGURE 3. The characteristics of the product developed

4.2. **The most used mathematics courses and their coverage in the curricula.** In this section, note that the usage of these courses also might reveal their importance in the software industry.

Participants were asked to provide how much they use the listed (i.e. five) mathematics courses in their professional careers (e.g., from *Never(0)* to *Always(5)*). According to the results, 'Probability & Statistics' is the most used mathematical-related course is with 1.892 rating value, followed by '(Propositional/Predicate) Logic'. 'Differential Equations' is the least used mathematics course (e.g., with 0.404 usage rating value over 5.0) as depicted in Figure 4. Note that the usage rating values are computed as an average of the rating scales.

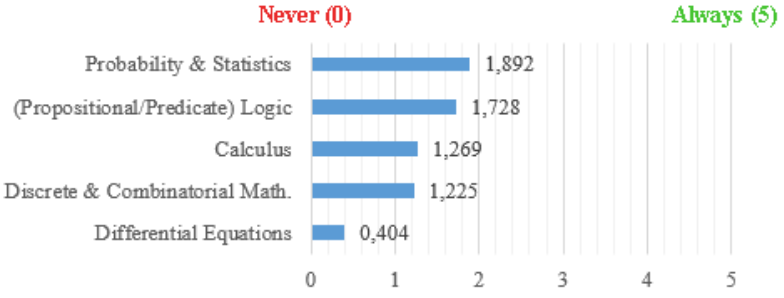


FIGURE 4. Usage of Mathematics Courses

To understand the coverage of these courses in the curriculum, the participants were asked whether they took such a course during the university education. As seen from Figure 5, all respondents took 'Calculus' and 'Probability & Statistics' courses (100%); whereas, 94.9% of respondents took 'Differential Equations', 86.12% of participants took '(Propositional/Predicate) Logic', and 71.73% of participants took 'Discrete & Combinatorial Mathematics' courses.

The results in Table 1 presents the coverage of mathematics-related courses depending on the computing discipline criteria. The coverage for 'Differential Equations' is greater in non-computing disciplines; on the other hand, 'Logic' and 'Discrete & Combinatorial Math' coverage is (significantly) greater in computing disciplines, which might be expected since their usage within the related curriculum is pre-requisite for other courses (e.g., in EEE curriculum, for signal processing courses (not mandatory for all computing disciplines), differential equations course is almost

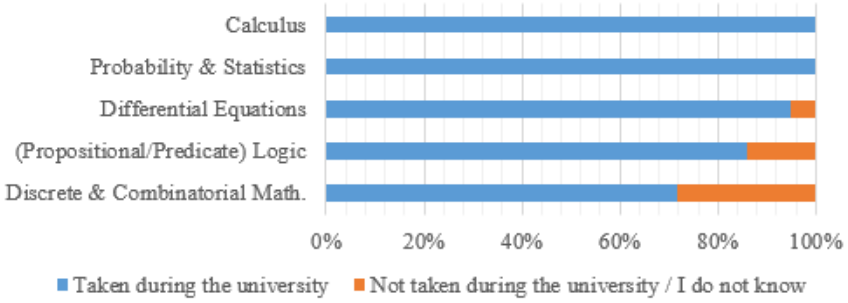


FIGURE 5. Usage of Mathematics Courses

TABLE 1. Mathematics-related topics coverage with respect to non-computing vs computing disciplines

Mathematics Courses	in non-computing discipline (122 responses)	in computing discipline (506 responses)	**overall coverage** (628 responses)	**usage rating value** over 5.0
Calculus	100%	100%	100%	1.269
Probability & Statistics	100%	100%	100%	1.892
Differential Equations	100%	93.67%	94.9%	0.404
(Propositional/Predicate) Logic	45.08%	95.85%	86.12%	1.728
Discrete & Combinatorial Math	8.19%	86.75%	71.73%	1.225

a 'must'); or in CS curriculum, for artificial intelligence (not included in most non-computing curriculum), discrete mathematics is a required course).

The knowledge gaps are shaped by the relation between the usage of a skill in the industry and the coverage of this skill in the curriculum. In other words, the highest usage ranking with the smallest coverage makes a skill (in our case, a mathematics course) have the greatest knowledge gap; whereas the highest usage ranking with the possible greatest coverage (or lowest usage ranking with the lowest coverage) might not result in a gap. According to the results, the greatest knowledge gap (e.g., negative gap) is in 'Differential Equations' course, which was taken by almost all participants (e.g., 94.9%) with the lowest usage rating (e.g., 0.404).

Note that, there is no courses, whose usage rating is higher than 2. This might denote that there is a possible misalignment of what is used in the industry and given in the academia (e.g., how it is taught) according to the perceptions of software practitioners. We can say that mathematics (as a tool for reasoning and thinking) should be included in curriculum (e.g., for some topics such as artificial intelligence, signal processing), but perhaps the weight (e.g., the given course hours) of some mathematics-related courses might be updated as in the case of 'Differential Equations'.

Our results are similar to what were investigated (see Section 3 for related work), which revealed that much of the mathematics knowledge was being forgotten, whereas much new software knowledge was being acquired in the industry. For example, Lethbridge suggested to give less importance to continuous mathematics (e.g., differential equations) in the curriculum [9], but our results still showed that making the changes in the curriculum is not as easy task as reporting the need for the changes.

As survey results showed that the importance and usage of mathematical skills and knowledge are purely dependent on various characteristics, we investigated some further cross-factor analysis:

- The participants, who use 'Calculus' at *Frequently(3)* and *Most of time(4)* degree, were non-computing graduates. In our data pool, there is no *Always(5)* usage for 'Calculus'.
- Software Testers is the SE role, which uses 'Probability & Statistics' at most with 3.276 usage rating value.
- The results showed that the participants, who uses 'AI & Machine Learning' concepts at least *Sometimes(2)* degree (e.g., from *Sometimes(2)* to *Always(5)*), uses 'Probability & Statistics' at least *Rarely(1)* and 'Discrete & Combinatorial Math' at least *Sometimes(2)*, denoting that for artificial intelligence this mathematics-related courses are necessary (e.g., there is no *Never(0)* usage).

- '(Propositional/Predicate) Logic' is the only mathematics course, which is used to some degree for all software developers and software architects (e.g., from *Rarely(1)* to *Always(5)*; there is no *Never(0)*).
- Software testers use '(Propositional/Predicate) Logic' at least *Sometimes(2)*; moreover, the participants, who rated the usage of logic concepts as *Always(5)* was working as software testing roles.
- There is no *Always(5)* usage for 'Differential equations', as 'Calculus'.
- The common characteristics of the participants, who use 'Differential equations', is that they work on the embedded applications (its usage value is 0.58 in embedded; whereas 0.04 in web applications; 0.09 in desktop application). Systems engineers in the embedded domain used it with 1.623 rating value (e.g., the SE role, who uses it at most).
- The participants, who did not use any 'Object-oriented (OO) concepts' (17.8% of the responses) use 'Differential equations' with 1.253 usage rating value.

Note that in our survey data, we have various SE knowledge areas (e.g., software requirement, design, and implementation, etc.) besides different SE Topics (e.g., algorithms, data structures, object oriented, artificial intelligence, etc.), which we studied their importance in the software industry [5]. For further cross-factor analysis related to Mathematical Foundations, the readers will be able to evaluate the results with the raw data [4].

4.3. **The opinions about decreasing the weight of mathematics related courses in curricula.** One of survey sections asks software practitioners about their suggestions to academic on various open arguments [5]. In this section, there is a mathematics-related argument, which proposes "*The weight of mathematics related courses (e.g., mathematical foundations) should be decreased in curricula*". The participants rated this argument from *Strongly disagree (-2)* to *Strongly agree (2)*. According to overall results, 117 participants (18.6%) strongly agree and 170 participants (27.1%) agree with this argument (e.g., 46% of the participants are positive for this proposal). On the other hand, 127 participants disagree and 54 participants strongly disagree with this argument (e.g., 29% of the participants are negative for this proposal). 25% of the participants are neutral to decrease the weight of mathematics-related courses in the curriculum. It is interesting that some practitioners, who rated their overall mathematics usage (e.g., the arithmetic average of 5 listed courses) below 1.0 rating, might be disagree with this argument since the effect of mathematical skills in software engineering is very important even if they do not directly use any output of mathematics courses in their current workforce.

When we examine the effect of highest academic degree on the overall results as a further cross-factor analysis, we see that the importance level of 'the weight of mathematics related courses' increased whenever the academic degree of the participants increased (e.g., from BSc to PhD) as depicted Figure 6.

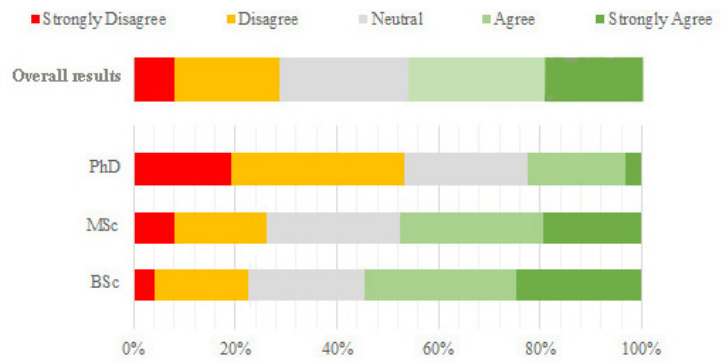


FIGURE 6. Opinions of practitioners depending on highest academic degree

Note that the correlation between the given importance (e.g., disagreeing with this argument) and the highest academic degree is statistically significant since their p-values¡0.001 [4].

We also analyzed the effect of product characteristics developed by the participants on the proposed argument as depicted in Figure 7; however we could not find a significant correlation as in the case of highest academic degree. The results showed that 'Finance & Banking' sector is the only sector, whose more than 50% of their participants

are negative on this argument (e.g., either disagree or strongly disagree). On the other hand, all the participants, who contributed to the survey from 'Government', strongly agree with this argument.

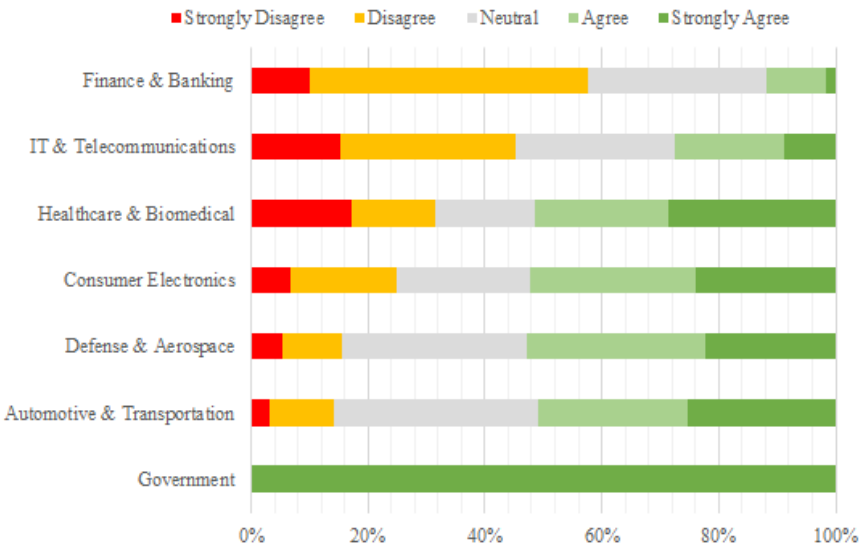


FIGURE 7. Opinions of practitioners depending on the industrial sectors

5. DISCUSSION

In order to bridge the skills hap in the software industry, we believe that the academia should teach how to lifelong learn. Our cross-factor analysis for any SE Topics (e.g., artificial intelligence, web programming, embedded software, even blockchain concepts) showed that there is no single curriculum, which can prepare all graduates for all roles in different context during such a technological era; hence university programs should focus slightly more on teaching 'how' to learn anything from anywhere so that practitioners can continue to learn new technologies as they encountered in the workplace. Since the technology is always changing and so are the importance of new SE topics, the author believed that the good software engineer is the person, who does not have to know everything, but must know where to find anything from anywhere to convert data into information and knowledge. To achieve this goal, mathematical skills is crucial for any software engineer by teaching them various forms of reasoning and abstract thinking skills even if they never directly use any mathematics courses learnt at the university. Abstraction, which is the core of software, is not only necessary for software developers but also requirement engineers. The main benefit any SE role (e.g., developer, systems engineers, testers etc.) got from the mathematics they learned at university was the experience of rigorous reasoning with purely abstract objects and solutions.

As our cross-factor results showed that the usage of some mathematics courses are directly related with the profiles of the participants and the characteristics of the product that practitioner developed. Although none of the overall usage rating of mathematical related courses is higher than 2 (i.e., less than *Sometimes(2)*), depending on SE role (e.g., software testers and systems engineers) and the SE topics used in the product (e.g., machine learning), the usage of mathematics courses increases as presented in Section 4.2. In other words, how necessary these courses are depends on what the software professionals are doing. For standard web applications, since software developer more focuses on user interface, networking, etc., which are all challenging in different ways themselves, but are less directly tied to mathematics. On the other hand, for more algorithmic software (even for web, but mostly in embedded and desktop applications), solving a more complex problem using graph theory, search algorithms, trees, etc., these courses are more important with respect to mathematical skills. At the other extreme, the fields such as machine learning, graphics, robotics, signal processing etc. might require mathematics knowledge from calculus, probability & statistics, logic, differential equations and various discrete mathematics topics.

Although the interpretation of the word 'use of mathematics courses' might vary among practitioners, we believe that the mathematics courses strengthen the process of learning and problem-solving skills even if there may not be much directly transferable information between these courses and typical software engineering activities. We presented similar perceptions in Section 4.3. Accordingly, 46% of the participants (e.g., less than half) agree that the weight of mathematics courses in the university education should be decreased. Concerning about the first investigation about 'the usage of these courses in the workplace' (e.g., none of their usage is more than *Sometimes(2)*), this results showed that some software professionals know about the necessity of mathematics courses even if they do not directly use them in their software products developed. Note that even in these data science days, where it is not strictly necessary (with the help of already built-in libraries), mathematical skills help increase the understanding of mathematical and statistical concepts without blindly accepting the output of software libraries.

In the light of these findings, our results showed that the usage rating given to mathematics courses software industry is very low compared to other technical skills (see [4] for various rating values for hard and soft skills); however, as our cross-factor analysis showed that mathematics (as a tool for reasoning and thinking) is still very important (used) and should be included in SE-related curriculum for various SE topics (e.g., artificial intelligence, signal processing) and SE roles (e.g., testers). Therefore, by considering the relation of specific mathematics courses to software engineering, the emphasis on certain mathematics courses should be updated. While modifying the curriculum, we believe that due to rapid changes in technology and industrial needs, this adaption should be forward-looking with a fast response to changes while supporting current practices in the software industry. To achieve this goal, industry and academia should work together in creating an up-to-date curriculum so that the industrial expectations will be fully met via mutual understanding.

6. Limitations and Threats to Validity

We minimized or mitigated the possible validity concerns according to the reference checklist [12].

To cope with construct validities, which is related to whether this survey investigated the different perceptions of software practitioners about mathematical skills in the software industry [12]. We collected data from different sources in order to avoid mono-operation bias.

In our survey design, one of the important decisions is the selection of the mathematical courses. We analyzed various SE-related BSc curriculum (not only computing disciplines but also EEE curriculum as well) in various (mostly in Turkish) universities during the design of the survey [3].

While mitigating external validity concerns [13], we distributed the survey to different software practitioners to decrease the effect of possible dominant participant number. Note that after preliminary analysis [1], we decided to include the responses of we only included the responses from 13 countries, where the participants completed their undergraduate degree (i.e., BSc) in Turkey. Therefore, the results are based on the analysis of data consisting of the perceptions of the Turkey-educated software practitioners. We cannot claim that the same results might be obtained in other parts of the world since our results for coverage of mathematical skills in the curriculum are mostly based on what Turkish universities would be teaching in software-related disciplines. Note that the majority of respondents are from Turkey (58%), which may have led to bias the results. To address this, we reported demographics of the participants and the readers will be able to evaluate the applicability in different contexts with the raw data [4].

By using pilot studies prior to the execution, we also improved the reliability of our survey [13].

7. Conclusion

In this study, different perceptions about the usage of the mathematics courses in the software industry with their coverage in the undergraduate programs in Turkish universities were investigated via a practitioner survey. We also presented the opinions of software practitioners about the current weight of these courses in the curriculum by proposing an open argument.

According to the results, considering the usage rating of mathematical related courses, we observed a possible misalignment of what is used in the industry and given in the academia (e.g., how it is taught). As survey results showed that the usage of mathematical skills (related with mathematics courses) are purely dependent on various characteristics such as profiles of the participants or the software product developed by that practitioner as we investigated with various cross-factor analysis. However, the interpretation of the word 'use of mathematics courses' might vary among practitioners; hence we believe that the mathematics courses strengthen the process of learning and problem-solving

skills even if there may not be much 'directly' transferable information between these courses and typical software engineering activities.

After analyzing the coverage and the usage rating of these courses, we can say that mathematics (as a tool for reasoning and thinking) should be included in SE-related curriculum, but perhaps the weight of some mathematics-related courses might be updated as in the case of 'Differential Equations'. While updating the curriculum, we believe that due to rapid changes in technology and industrial needs, this adaption might be achieved via collaboration between industry and academia. However, note that there might be some challenges during this adaption due to some accreditation programs (such as ABET), which might prevent some curriculum changes. Therefore, such possible limitations might be analyzed as a future direction to better understand the difficulties in the academic side.

The results also showed that more than half of the participants (54%) disagree with the argument, which proposes to decrease the weight of mathematics courses in the academic curriculum even if they do not directly use them in their software products developed. This finding also revealed that software professionals are aware of the necessity of mathematical skills, but concerning about their 'direct' usage, they might expect some curriculum changes. Note also that some software engineering related topics such as Cryptography may include heavy mathematics materials. As future work, it will be beneficial to investigate the relation of these SE courses and given-mathematics courses to better design the corresponding curriculum.

Acknowledgement

The author would like to thank all software practitioners, who contributed to this survey.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] Akdur, D., *A Survey on Bridging the Gap between Software Industry and Academia: Preliminary Results*, in 13th Turkish National Software Engineering Symposium, 2019, https://www.researchgate.net/publication/334559697_A_Survey_on_Bridging_the_Gap_between_Software_Industry_and_Academia_Preliminary_Results. 6

[2] Akdur, D, Modeling Patterns and Cultures of Embedded Software Development Projects, Thesis, Doctor of Philosophy (PhD), Information Systems, Middle East Technical University (METU), 2018. 1

[3] Akdur, D., *The Design of a Survey on Bridging the Gap between Software Industry Expectations and Academia*, in 8th Mediterranean Conference on Embedded Computing (MECO), Montenegro: IEEE, 2019, doi:10.1109/MECO.2019.8760101. 1, 2, 6

[4] Akdur, D., Raw Data Results: Survey on Bridging the Skills Gap, Mendeley Data, v1, http://dx.doi.org/10.17632/fx537tpw6f.1, 2020. 4.2, 4.3, 5, 6

[5] Akdur, D., *Survey Form: Bridging the Gap between Software Industry Expectations and Academic Activities*. Available: https://drive.google.com/file/d/1-1DAexMmd4_om_kahwDXLBwYdCYJ_l2m/view, (accessed Mar 15, 2019). 2, 4.2, 4.3

[6] Bourque, P., Fairley, R. E., *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*: IEEE Computer Society, 2014. 1

[7] Devlin, K. J., The math gene: How mathematical thinking evolved and why numbers are like gossip: Basic Books, 2001. 1

[8] Kitchenham B., Budgen D., Brereton P., Woodall P., *An investigation of software engineering curricula*, Journal of Systems and Software, **74**(3)(2005, 325-335, doi:https://doi.org/10.1016/j.jss.2004.03.016. 3

[9] Lethbridge T. C., *A survey of the relevance of computer science and software engineering education*, in Proceedings 11th Conference on Software Engineering Education, 1998, doi:10.1109/CSEE.1998.658300. 3, 4.2

[10] Linåker J., Sulaman S.M., Maiani de Mello R., Höst M., Runeson P., Guidelines for Conducting Surveys in Software Engineering, 2015. 2

[11] Shull F., Singer J., and Sjøberg D. I. K., Guide to Advanced Empirical Software Engineering, Springer-London, Inc., 2010. 2

[12] Surakka S., *What subjects and skills are important for software developers?*, Commun. ACM, **50**(1)(2007), 73-78, doi:10.1145/1188913.1188920. 3, 6

[13] Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B., Wesslén A., Experimentation in Software Engineering, Springer Berlin Heidelberg, 2012. 6