Journal of Information Technology Education: Research

An Official Publication of the Informing Science Institute
InformingScience.org

JITEResearch.org

Volume 19, 2020

INFORMATION TECHNOLOGY ASYMMETRY AND GAPS BETWEEN HIGHER EDUCATION INSTITUTIONS AND INDUSTRY

Yasar Guneri Sahin *	Izmir University of Economics, Izmir, Turkey	yasar.sahin@ieu.edu.tr
Ufuk Celikkan	Izmir University of Economics, Izmir, Turkey	ufuk.celikkan@ieu.edu.tr

* Corresponding author

ABSTRACT

Aim/Purpose	This paper investigates the gaps between industry and academia perceptions of information technology fields, such as computer science, software engineering, and computer engineering, and it identifies areas of asymmetry between curricula and industry expectations. The study mainly focuses on the skills required of IT professionals (graduated students) and on how higher education institutes equip students for industry.
Background	Higher education institutes have several IT-related departments. However, it is not clear whether these departments have sufficient content to equip students with industry-related skills. Rapid advances mean that some curriculum topics are redundant before the end of a standard two- or four-year degree programs. Balancing the technical/non-technical skills and adjusting the curricula to better prepare the students for industry is a constant demand for higher education institutions. Several studies have demonstrated that a generic curriculum is inadequate to address current IT industry needs.
Methodology	The study involved a comprehensive survey of IT professionals and companies using a Web-based questionnaire sent directly to individual companies, academics, and employers. 64 universities and 38 companies in 24 countries were represented by the 209 participants, of whom 99 were IT professionals, 72 academics, and 38 employers.
Contribution	This paper is intended to guide academics in preparing dynamic curricula that can be easily adapted to current industry trends and technological develop-

Accepting Editor Keith A. Willoughby │Received: February 21, 2020│ Revised: April 24, 2020 │ Accepted: May 2, 2020.
Cite as: Sahin, Y. G., & Celikkan, U. (2020). Information technology asymmetry and gaps between higher education institutions and industry. *Journal of Information Technology Education: Research, 19,* 339-365. https://doi.org/10.28945/4553

ments, with content directly relevant to student's careers. In addition, the results may identify the skills that students need to secure employment and the courses that will provide skills in line with current industry trends.

Findings: The results indicate a lack of emphasis on personal and non-technical skills in undergraduate education compared to general computer science, software development, and coding courses. Employers' and software experts' responses emphasize that soft skills should not be ignored, and that, of these, analytical thinking and teamwork are the two most requested. Rather than a theoretical emphasis, courses should include hands-on projects. Rapid developments and innovations in information technologies demand that spiral and waterfall models are replaced with emerging software development models, such as Agile and Scrum development.

Recommendations for Practitioners: A multidisciplinary approach should be taken to the teaching of soft skills, such as communication, ethics, leadership, and customer relations. Establishing multiple learning tracks in IT education would equip students with specialized knowledge and skills in IT. An effective communication channel should be established between students and industry. It is also important to reduce the distance between academics and students and to provide an interactive environment for technical discussions. Enterprise level computing and Framework use provide job market advantages.

Recommendations for Researchers: Researchers and department heads, particularly those involved in curriculum design and accreditation, could use the results of this exemplary study to identify key topics for attention.

Impact on Society: Changes of various degrees are required in the current curricula in many higher education institutions to better meet student needs. Societies and technology are dynamic in nature, and information technology-related curricula in higher education institutions should be equally dynamic.

Future Research: Since technology (especially information technology) transforms and advances itself so rapidly, this study should be replicated t to investigate how these changes affect the gap between revised curricula and current industry expectations.

Keywords: asymmetry in software education, higher education and industry gaps, information technology education

INTRODUCTION

Information technology (IT) has become an indispensable part of industry and business markets and of daily life. Furthermore, as an important coefficient of countries' development evaluation formula, IT is playing a greater role in living standards and welfare levels (Anderson & Rainie, 2018). Increase in use of social media, mobile devices, and the Internet (Arellano & Cámara, 2017) has led to many new technology companies, increasing competition. Hence, an increase in the number of IT companies has triggered a new challenge, i.e., finding well-qualified professionals. There is another challenge caused by rapid development of technology, that is, volatility in the required professional skills (Celikkan & Sahin, 2017). Hence, educating IT professionals has become even more challenging. Since education is the most important part of improving skills of professionals, institutions should try to create more flexible and dynamic IT programs to adapt their curricula to emergent changes in technology. Moreover, technical knowledge alone is insufficient; business operations, presentation, and social skills are also inevitable parts of this professional education.

In the new era, most higher education institutes have several IT related departments, such as Information Technology, Computer Science, Computer Engineering or Software Engineering, Computer and Electrical Engineering, and, sometimes, computer programming, industrial design, and hardware and software related technical programs. However, it is not clear whether these departments have content able to fully equip students with industry-related skills (Benamati et al., 2010; Föll et al., 2018; Gonzalez-Torres, 2018; Sahin & Celikkan, 2017; Wang & Li, 2000). From a quantitative perspective, the rapid advances in IT industry make some curriculum topics redundant before the end of a standard two- or four-years degree programs. At this point, a particularly important question is whether academia can equip students with the necessary skills. Moreover, the following questions need answers:

- What are the skills that the IT market expects from new graduates?
- What are the skills that the academics believe the new graduates possess?
- Do academics teach what the IT market requires and produce competent IT professionals?
- Is the IT market excessively concerned with profession- related skills and ignoring others such as soft skills?
- How one can achieve a high correlation between the information delivered, and knowledge absorbed by the student?

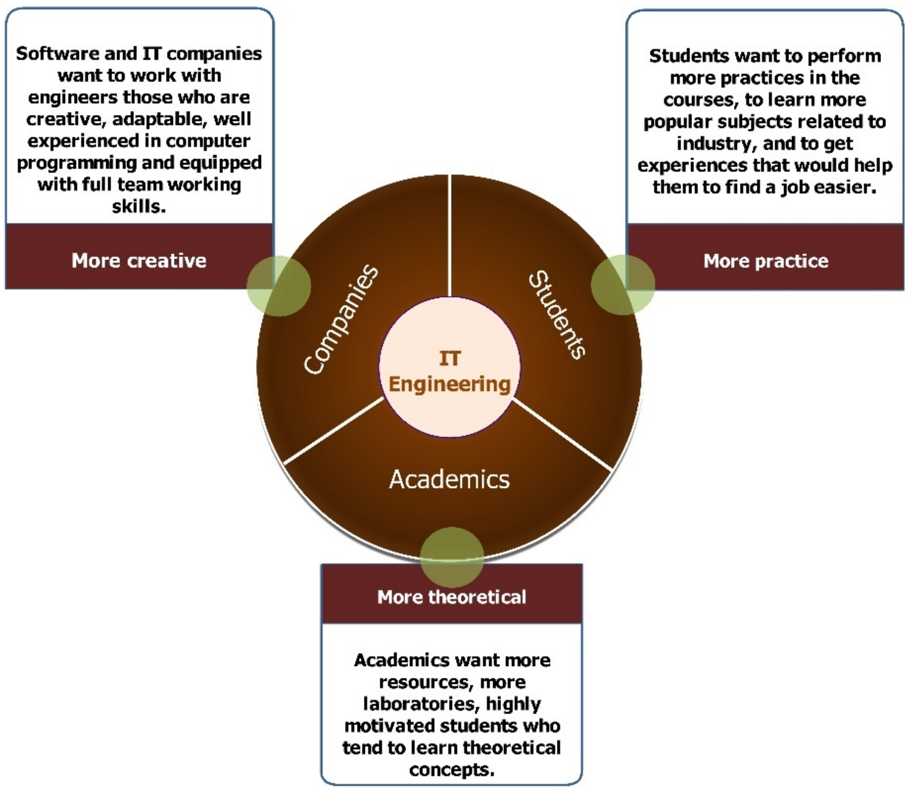

Figure 1. The expectations of three stakeholders of the IT sector

These questions have different target audiences and are addressed to different IT stakeholders, depending on their expectations. We have identified and targeted three stakeholders in this paper: (1) Students, (2) IT Companies and (3) Academics. Figure 1 illustrates generic IT expectations of these three stakeholders.

This study intends to address the questions presented above and to reveal the actual needs of the Software Development Industry (SDI) and which skills new IT professionals or candidates have upon graduating. Existing research into the skills required by IT professionals includes number of studies conducted to shed light on the IT skills needed for a successful career, the skills acquired by graduates, and the appropriate curricula to be followed. This study additionally discusses how the asymmetry between the academia and industry can be addressed, and how academia can adapt to emergent technologies.

LITERATURE REVIEW

The term “software engineering” emerged in 1968, and since then the field has greatly progressed, as reported in (Mead, 2009). A very early workshop discussion focused on the interrelation problems between hardware and software in computer science education, in which Dennis (1969) presented a three-subject solution in computer systems and programming, emphasizing the importance of understanding the relationship between computer hardware organization and the implementations. Wasserman and Freeman (1978) described the problems in training software engineers and made recommendations for the future of education in the field. Many studies have since tried to evaluate existing computer science and software engineering education in comparison to industry needs.

In a survey, Lethbridge (1998, 2000) aimed to assess the relevance of computer science and software engineering education. The study concluded that many students would prefer engineering related courses to mathematics courses in the curricula.

Another study conducted by Kitchenham et al. (2005) investigated the relations and matches between deliveries in four UK universities and the requirements of the UK software industry. Although relevant, it is limited to four UK universities and, therefore, may not reflect a truly global view.

Penzenstadler et al. (2013) stated that, although the importance of case study-based education is increasing, some teachers persist with traditional software engineering teaching methods. In addition, Penzenstadler et al. emphasized the significance of working with real world stakeholders and industrial partners during software engineering education. Hayes (2002) investigated how industrial projects can enhance the theoretical knowledge acquired during software engineering education, highlighting the benefits of applied projects. Preston (2004) suggested that IT students learn better during authentic team projects for real-world customers, and Callahan and Pedigo (2002) asserted that curricula should create balance by incorporating realistic problem-solving exercises into theoretical material.

Skevoulis (2011) and Beckman et al. (1997) suggested that universities establish separate programs in collaboration with industry to meet organizations’ software-related education and training needs. Such collaborations and programs help adjust the undergraduate curricula to satisfy the needs of a high-tech, global marketplace. The direct industry involvement in curriculum design was also shown to be beneficial by Carrington et al. (2005) and Pilgrim (2013), which also emphasized the tension between the university’s goals of broader education for life-long learning, and industry’s need for particular skills.

Several studies have demonstrated that technical skills alone are insufficient for a successful IT career (Aasheim et. al., 2019; Lee et al., 1995). Lee et al. (1995) found that IT professionals need a broad technical management background in addition to in-depth knowledge of specific technologies. As well as technology, a substantial knowledge of business and human relations is needed; therefore, a generic curriculum is inadequate to address the needs of IT industry. The study recommends teaching business and organizational management to technology-oriented students, and information technology to executives and managers. Burnell et al. (2002) has similar findings, i.e., teaching standard software engineering curriculum in isolation from other disciplines fails to prepare students sufficiently for the software product development environment. The courses need to include multidisciplinary and, possibly, multi-institutional projects to expose the students to teamwork and enhance communication. Plaza et al. (2013) argue that an innovative and continuous improvement methodology improves the execution of a course, and their experiments showed that these improvements increased motivation and enrollment. In essence, incorporation of new technological trends and integrating research and innovation into classroom has a positive impact on the students.

Morena et al. (2012) investigate whether the knowledge taught in Software Engineering Curricula satisfies industry expectations by comparing the graduate and undergraduate Software Engineering curriculum guidelines to typical job profiles on Career Space websites. European Union has created the Career Space initiative, geared towards to more experienced IT professionals, to address the European shortage of "people with a fluent digital age language." The finding is that there needs to be a move from technical to business topics in Software Engineering education. IT business consultancy and soft skills represent the areas of greatest mismatch between the job profiles of Career Space and Software Engineering education. Kim et al. (2006) surveyed a small number of managerial and non-managerial employees with both IS and non-IS focus, also finding that project management, personal productivity tools, soft skills, and security are perceived to be important. Radermacher et al.'s (2014) research with employers and hiring managers found that lack of project experience and problem-solving abilities were the primary obstacles to graduates' employment.

Other studies also reached similar conclusions, that non-technical skills are paid insufficient attention in tertiary education. Cappel's (2002) study investigated the employer's view of new Information System graduates' "expected" and "actual" performance on technical and non-technical skills. The gaps were greatest for non-technical skills, such as oral and written communication, problem solving, and ability to learn. Similar conclusions were reached by Callahan & Pedigo (2002), who focused on business executives' perceptions of the skills lacking in potential candidates and priorities for university teaching. IT professionals lack technical management skills rather than software programming and hardware skills, and the industry needs information engineers capable of designing business processes, while understanding that IT is the most essential element of a business (Brynjolfsson & Saunders, 2010). In a similar study, Tesch et al. (2008) focused on employers' perceptions of gaps in personal and interpersonal skills, similarly finding that that the areas of teamwork, listening skills, and verbal communication represent the greatest discrepancy between practitioners' expectations and new hires' skill delivery. Plice & Reinig's (2007) study also shows that, in the long run, education can best support career development by emphasizing communication and teamwork skills, while maintaining a balance between technical and business skills. Medlin et al.'s (2001) study reports that communication, analytical, and managerial skills are important, while Merhout et al.'s (2009) qualitative analysis, based on focus group discussions of their academic advisory board, concludes that soft skills account for seven of the top ten critical skills sought by information technology executives.

Material and Method

Survey Method

The survey was conducted using a Web-based questionnaire publicized by directly approaching companies (employers), academics, and professionals. Participants from the IT related fields were randomly selected from LinkedIn, ResearchGate, and Google Scholar, and the survey link was sent to all potential participants via email, LinkedIn, and ResearchGate. The average questionnaire completion time was 21 minutes. The average was lengthened by the written comments of many participants, which provided excellent insights.

The primary intention of the study is to determine the knowledge gap and, therefore, a curriculum to maximize the satisfaction of industry requirements and to identify student and teacher expectations. The study, therefore, consulted three stakeholder groups: software professionals, employers, and academics. Separate questionnaires for each stakeholder group were designed, each consisting of two parts. The first part, covering demographic questions, varied among the groups, but the second part was remarkably similar for all three groups and intended to evaluate the technical and non-technical IT skill competences. The competence areas are based on topics from the IT profession under three general categories: computer, software, and general skills. Profession-based questions were prepared after a comprehensive investigation into the current curricula of computer science, computer, and software engineering departments of many different universities and IT companies from several different countries and SWEBOK (Software Engineering Body of Knowledge, https://www.computer.org/web/swebok). This body of knowledge is recognized by IT practitioners and used by curriculum developers. The questions focus on 71 topics, organized into six categories, encompassing a wide variety of skill and knowledge topics. The taxonomy is based on IT technology and processes. No attempt has been made to distinguish between concepts such as knowledge versus skills or education versus training. The process of mapping of these constructs onto the job functions and requirements is complex and often obscure. Moreover, for pedagogical and theoretical reasons, the pragmatic nature of businesses and IT tasks require a combination of education and skills with no clear differentiation between them (Lee et al., 1995).

The software professionals were asked to rate (a) the extent to which their formal education developed their skills and (b) the level of skill needed in their jobs for each of these 71 topics, using a five-point Likert scale, with 1 indicating formal education has very low impact on their skill development, and 5 a very high impact. The employers were asked to rate their skill expectation from IT professionals on the same 71 topics using the same scale, with 1 = (very low expectation), and 5 = (excellent expectation) as shown in the following ordinal scale:

Score	Software Professional	Employer
1	Very Low Impact	Very Low Expectation
2	Low Impact	Low Expectation
3	Medium Impact	Medium Expectation
4	High Impact	High Expectation
5	Excellent Impact	Excellent Expectation

The academics were asked whether they taught each of these 71 topics, and responses were normalized to a numeric score, based on the following scale:

Score	Academics
0	No idea
0	We do not teach
4	We teach
5	We teach with hands-on examples

In this part of the survey academics evaluated their teaching style for their courses. The academics are qualified to teach these courses, and it is assumed that they have the desired level of knowledge; therefore, the survey is only concerned with two issues: whether a subject is being taught by the academics and the type of teaching (theoretical vs practical, i.e., with hands-on examples). The survey does not seek any data on the levels of theoretical or practical teaching. It is presumed that academics who lack knowledge on a subject are not expected to teach that topic, and a value of is 0 is used as the answer if the course is not taught. If they teach a subject theoretically, a value of 4 is assigned (a high level of knowledge transferred to students), and if they use hands-on examples (reinforced knowledge transfer), a value of 5 (a very high level of knowledge transfer). Accordingly, the results were normalized to the values shown in the scales above.

THE PARTICIPANTS

In this survey, 209 participants represented 64 universities and 38 companies in 24 countries. Of 209 participants, 99 were IT professionals, 72 academics and 38 employers. Tables 1, 2 and 3 show the demographic details of the participants.

The majority of the employers were from national companies (defined as operating within the borders of a particular country) with a single center of operation (78%). 55% of the companies surveyed had 10 or fewer employees, while 13% had 50 or more. 63% of companies were in the IT business. Table 1 shows the employer demography.

Table 1. Employer demographic data

Description	# of Participants n=38	Percentage %
The type of the company		
National company[1]	30	78.95
National company with more than one branches	2	5.26
International company[2]	4	10.53
Other[3]	2	5.26
The number of employees		
1-10	21	55.26
11-50	12	31.58
51-250	3	7.89
251-1.000	1	2.63
1.001-10.000	0	0.00
More than 10.000	1	2.63
Core business of the companies		
General Information Technology Market (Software/Hardware)	24	63.16
Telecommunication	3	7.89
Consultancy	3	7.89
Military/Defense Industry	2	5.26
Transportation	1	2.63
Academics	1	2.63
Banking	0	0.00
Medical	0	0.00
Media	0	0.00
Other	4	10.53
Preferred graduated programs of professionals to recruit		
Computer Engineering	35	92.11
Software Engineering	28	73.68
Computer Programming	17	44.74
Computer Science	16	42.11
Computer. Education and Instructional Technologies	8	21.05
Computer Technologies and Education	5	13.16
Other	6	15.79

[1] National company (business) is the one that operates within the borders of a home country.

[2] International company (multinational corporation) has facilities and other assets in at least one country other than its home country.

[3] Undisclosed

Table 2 shows the IT professionals' demographic data. Sixty Five percent of the software professionals had a bachelor's degree, and the remainder had postgraduate degrees (MS and Ph.D). Forty three percent of the software professionals had degrees in software engineering, 41% in computer engineering, and the remaining 6%, a degree in Computer Science. Three times more males than females participated in the survey. The overwhelming majority of software professionals received their education in English (88%), and 85% of the participants found English based education extremely beneficial. (The questionnaire allowed respondents to omit this question if not applicable. For this reason, the sum is not equal to 99). The group is relatively inexperienced, as 84% had five or fewer years of industry experience since graduation. The advantage of this was that the curriculum was unlikely to have changed greatly since their graduation. Eighty Eight percent (88%) of the software professionals had worked in three or fewer companies. Forty-four (44%) worked in general purpose IT companies, 14% in the banking industry and 12% is in the telecommunication sector.

Table 2. Software professional demographic data

Description	#of Participants n=99	Percentage %
Highest degree		
BSc (Bachelor of Science)	64	64.65
MS (Master of Science)	31	31.31
PhD (Philosophy of Doctorate)	4	4.04
Gender		
Female	24	24.24
Male	75	75.76
Graduated department/program		
Software Engineering	43	43.43
Computer Engineering	41	41.41
Computer Science	6	6.06
Computer Programming	2	2.02
Computer Technologies and Education	0	0.00
Computer. Education and Instructional Technologies	0	0.00
Other:	7	7.07
The medium of instruction		
English	87	87.88
Other:	12	12.12
Effects of the medium of instruction in a foreign language		
Extremely beneficial	37	43.02
Very beneficial	36	41.86
Moderately beneficial	6	6.98
Slightly beneficial	5	5.81
Not at all beneficial	2	2.33
Years past after graduation		
0- Not graduated yet	1	1.01
1	14	14.14
2	16	16.16
3	18	18.18
4	28	28.28
5	6	6.06
6	3	3.03
7	2	2.02
9	2	2.02
10-15	5	5.05
20+	4	4.04
How many different companies the professional worked for so far after graduation		
0- No job	2	2.02
1	38	38.38
2	32	32.32
3	15	15.15
4	7	7.07
5	2	2.02
6+	3	3.03

The core field (business) of the recent job		
General Information Technology Market (Software/ Hardware)	44	44.44
Banking	14	14.14
Telecommunication	12	12.12
Consultancy	7	7.07
Academics	6	6.06
Medical	3	3.03
Military/Defense Industry	1	1.01
Transportation	0	0.00
Media	0	0.00
Other (Mostly. e-commerce)	12	12.12

As shown in Table 3 the academics participating in the survey are concentrated in one of the following three disciplines (82%): computer science, computer engineering, and software engineering.

Table 3. Academics demographic data

Description	# of participants n=72	Percentage %
The main medium of instruction		
English	43	59.72
Other	29	40.28
Department/program		
Computer Science	31	43.06
Computer Engineering	15	20.83
Software Engineering	13	18.06
Computer Technologies and Education	1	1.39
Computer. Education and Instructional Technologies	0	0.00
Computer Programming	0	0.00
Other (Mostly CS&SE)	12	16.67
Position in institution		
Lecturer (Instructor, Professor, Dr.)	62	86.11
Head of Department	8	11.11
Dean	1	1.39
Other	1	1.39

The Survey Questions

The survey included 71 topics grouped under 6 categories: General Computer Science (10 topics), Development and Coding (7 topics), SW Project Phases, Processes and Management (17 topics), CASE Tools and Standards (6 topics), Model, Methodologies and Frameworks (19 topics), General Skills (12 topics). The software professionals were asked the following question to rate the skills they acquired during education and the relevance of their skills to their current job:

QUESTION: **"To what extend has your education developed your skills in the following subjects/fields, and to what extend do you need these at your job/s?"**

a) Learned this subject with this level during BS education
b) Need this subject with this level in my job

(1: Very Low; 2: Low; 3: Medium; 4: High; 5: Excellent)

Using the same 71 topics, the employers were asked the following question to rate their skill expectation from software professionals:

QUESTION: **"How would you rate your skill expectations from IT or software professionals about the following subjects/fields?"**

(0: N/A; 1: Very Low; 2: Low; 3: Medium; 4: High; 5: Excellent)

The academics were asked to indicate whether they taught any of these 71 topics, and if so whether they support teaching with practical examples.

QUESTION: **"Which of the following areas/subjects do you teach in your department/program?"**

(0: No Idea; 0: We don't teach; 4: We teach; 5: We teach with hands-on example).

Results and Discussions

In this section, the survey results from each of the three groups (professionals, academics, and employers) and related discussions, are presented separately.

IT Professionals (Education vs. Expectation)

Basic analysis of quantitative data was performed by calculating the mean values and standard deviation for each of the 71 topics evaluated by participants. T-tests were used to test for significant differences between means of knowledge acquired during education and the means of knowledge needed in work lives (de Winter & Dodou, 2010). Qualitative data was gathered through an open-ended question asking for comments on the education of software professionals.

Tables 4 and 5 show the software professionals' responses to the statements: "learned this subject during education" and "need this subject at my job." The difference between the answers to these pair of statements should be carefully assessed in relation to the level of skill/knowledge needed at work, in order to prioritize the curriculum adjustments efforts. Focusing on the differences themselves in isolation can be misleading, for example, a less-needed skill with a greater knowledge gap should have lower priority than a much-needed skill/topic with a smaller knowledge gap. A large knowledge gap for a highly rated skill indicates that the curriculum is insufficient in preparing the student for the IT industry.

Tables 4 and 5 clearly show meaningful differences (asymmetry) for certain topics, and differences are especially significant in the general skills area. This may be an indicator that as academics, we need to adjust our curricula towards more industry-integrated content in these areas.

Table 4. Software professional responses about computer science and software engineering-based courses and skills

Category/Subject/Field	Learned during education		Need in job		P (<0.05)	Meaningful Difference?	Increase / Decrease in Demand
	Mean	SD	Mean	SD			
General Computer Science							
Computer Networks	2.26	1.31	2.57	1.57	0.0853	No	↑
Data Communication	2.33	1.44	2.83	1.73	0.0011	Yes	↑
Data Structure and Algorithms	3.41	1.40	3.78	1.52	0.0403	Yes	↑
Logic Design	2.91	1.48	2.60	1.63	0.1398	No	↓
Database Systems	3.30	1.51	4.22	1.34	0.0000	Yes	↑
Computing Theory	2.38	1.56	1.86	1.41	0.0004	Yes	↓
Computer Hardware Courses	1.76	1.29	1.74	1.29	0.8926	No	↓
Operating Systems	2.52	1.51	2.74	1.57	0.2674	No	↑
Security (Web, Network, etc.)	1.98	1.35	2.91	1.66	0.0000	Yes	↑
Numerical Analysis	2.19	1.52	2.11	1.61	0.5761	No	↓
Development and Coding							
Concepts of Programming Lang.	3.45	1.42	3.97	1.53	0.0015	Yes	↑
Object Oriented Programming	3.27	1.54	3.94	1.57	0.0001	Yes	↑
More than one Programming Lang	3.08	1.54	3.63	1.60	0.0035	Yes	↑
Implementation Patterns	2.42	1.57	3.10	1.76	0.0001	Yes	↑
Embedded Systems Programming	1.44	1.13	1.41	1.20	0.8219	No	↓
Languages for Web Programming	2.30	1.45	3.27	1.75	0.0000	Yes	↑
Visual Design and Animation	1.30	0.97	1.64	1.35	0.0145	Yes	↑
SW Project Phases, Processes & Management							
Requirement gathering	2.62	1.77	2.99	1.88	0.0277	Yes	↑
Feasibility studies	2.41	1.66	2.87	1.82	0.0129	Yes	↑
Cost models	1.99	1.45	2.17	1.65	0.3242	No	↑
Business models	1.99	1.55	2.44	1.79	0.0088	Yes	↑
Requirement analysis	2.25	1.75	2.67	1.88	0.0084	Yes	↑
Design	2.64	1.55	3.44	1.75	0.0000	Yes	↑
Architecture	2.46	1.48	3.43	1.80	0.0000	Yes	↑
Test	2.26	1.49	3.51	1.64	0.0000	Yes	↑
Verification and Validation	2.23	1.53	3.05	1.77	0.0000	Yes	↑
Quality	2.07	1.49	2.90	1.79	0.0000	Yes	↑
Measurement	1.83	1.42	2.42	1.68	0.0002	Yes	↑
Maintenance	1.84	1.41	2.79	1.80	0.0000	Yes	↑
Reuse	2.22	1.58	3.24	1.79	0.0000	Yes	↑
Implementation	2.76	1.68	3.76	1.79	0.0000	Yes	↑
Configuration management	1.74	1.31	2.72	1.81	0.0000	Yes	↑
Risk management	1.71	1.31	2.31	1.77	0.0004	Yes	↑
Project management	2.15	1.42	2.74	1.75	0.0012	Yes	↑
CASE Tools and standards							
Generic CASE tools	2.38	1.68	2.44	1.78	0.7118	No	↑
UML	3.08	1.78	2.78	1.73	0.0645	No	↓
Documentation tools	2.43	1.60	2.95	1.76	0.0022	Yes	↑
Test tools	1.78	1.38	2.77	1.81	0.0000	Yes	↑
CMMI	1.62	1.38	1.80	1.56	0.2512	No	↑
Cost analysis tools	0.09	0.41	0.12	0.52	0.2589	No	↑

Table 5. Software professional responses about software engineering methodologies and approaches, and general skills

Category/Subject/Field	Learned during education		Need in jobs		p (<0.05)	Meaningful Difference?	Increase / Decrease in Demand
	Mean	SD	Mean	SD			
Model, Methodologies &Frameworks							
Agile Software Development	1.93	1.49	3.11	1.80	0.0000	Yes	↑
Aspect Oriented Software Eng.	1.38	1.27	1.83	1.63	0.0026	Yes	↑
Best Practice	1.66	1.36	2.49	1.89	0.0000	Yes	↑
Component Based Software Eng.	1.58	1.42	2.19	1.86	0.0001	Yes	↑
Constructionist design methodology	1.20	1.25	1.43	1.52	0.0472	Yes	↑
Design by Use	1.38	1.34	1.77	1.70	0.0024	Yes	↑
Design-Driven Development	1.49	1.37	1.90	1.72	0.0032	Yes	↑
Expletory Development	1.21	1.21	1.42	1.67	0.1333	No	↑
Extreme Programming	1.54	1.38	1.80	1.58	0.0657	No	↑
Object Oriented Software Eng.	2.82	1.78	3.38	1.94	0.0008	Yes	↑
Pair Programming	1.53	1.39	1.73	1.53	0.0746	No	↑
Iterative &Incremental Development	2.17	1.70	2.99	3.36	0.0093	Yes	↑
Rational Unified Process	1.28	1.30	1.28	1.40	1.0000	No	↓
Scrum management	1.43	1.29	2.38	1.87	0.0000	Yes	↑
Spiral model	1.76	1.54	1.64	1.47	0.3965	No	↓
Software Scouting	1.11	1.13	1.28	1.39	0.1005	No	↑
Test-driven development	1.76	1.42	2.53	1.83	0.0000	Yes	↑
Throw away prototyping	1.27	1.30	1.27	1.29	1.0000	No	↓
Waterfall Model	2.65	1.80	2.31	1.75	0.0635	No	↓
General Skills							
Customer relationships	1.60	1.30	3.05	1.84	0.0000	Yes	↑
Communication skills	2.26	1.57	3.75	1.80	0.0000	Yes	↑
Interview-Negotiation-Contract management	1.30	1.11	2.17	1.85	0.0000	Yes	↑
Team working	3.01	1.56	4.07	1.55	0.0000	Yes	↑
Marketing	1.16	0.93	1.79	1.55	0.0001	Yes	↑
Entrepreneurship	1.71	1.36	2.71	1.83	0.0000	Yes	↑
Ethical issues	3.06	1.81	3.31	1.89	0.1309	No	↑
Leadership	2.12	1.59	3.23	1.79	0.0000	Yes	↑
Presentation skills	3.05	1.70	3.58	1.74	0.0028	Yes	↑
Economy	2.27	1.60	2.01	1.66	0.1045	No	↓
Accounting	1.86	1.46	2.48	5.59	0.2490	No	↑
Analytic thinking	3.32	1.71	4.02	1.74	0.0000	Yes	↑

Figure 2 lists ten most common topics studied during education, and the ten most needed topics at work. Eight of the ten most common topics in formal software education also appear in the list of most needed at work, i.e., all except UML and Ethical issues. In place of these less important issues, respondents indicated that "Team Working" and "Implementation" are needed. Based on the mean differences of the respondents' answers, students acquired excessive knowledge on UML, but insufficient knowledge on ethical issues even though ethical issues dropped off from the list of ten most needed topics at work (Table 5 shows that there is still a knowledge gap in ethical issues). Respondents' answers indicate a need to focus on ethical issues, but these issues' relative importance has since declined compared to other more technical topics. The knowledge gap in "Team Working" and the "Communications Skills" topics, as shown in Table 5, indicates a need for improved interpersonal and management skills.

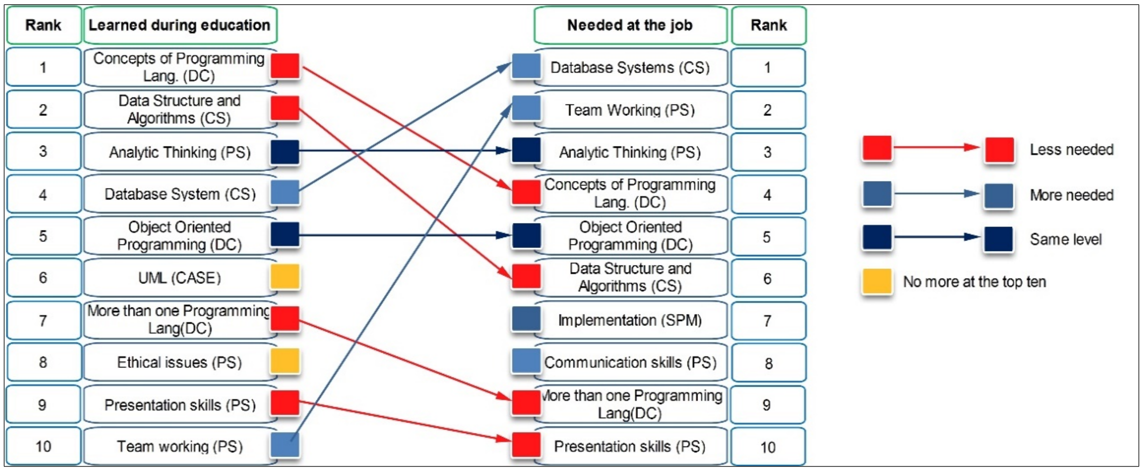

Figure 2. The asymmetry between education and workplace

Figures 3 and 4 show an executive summary of the knowledge gap in each category and the importance of topics in each category, based on the responses of Software Professionals.

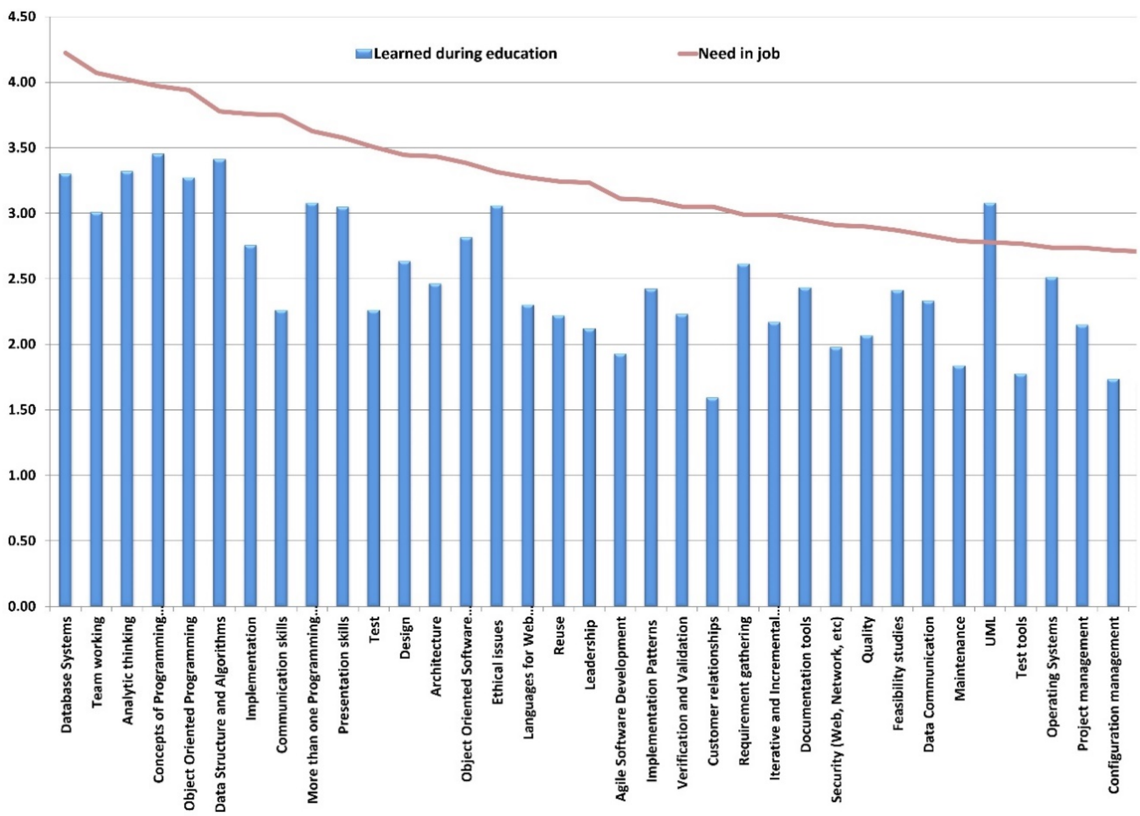

Figure 3. Executive summary of knowledge gap as reported by professionals

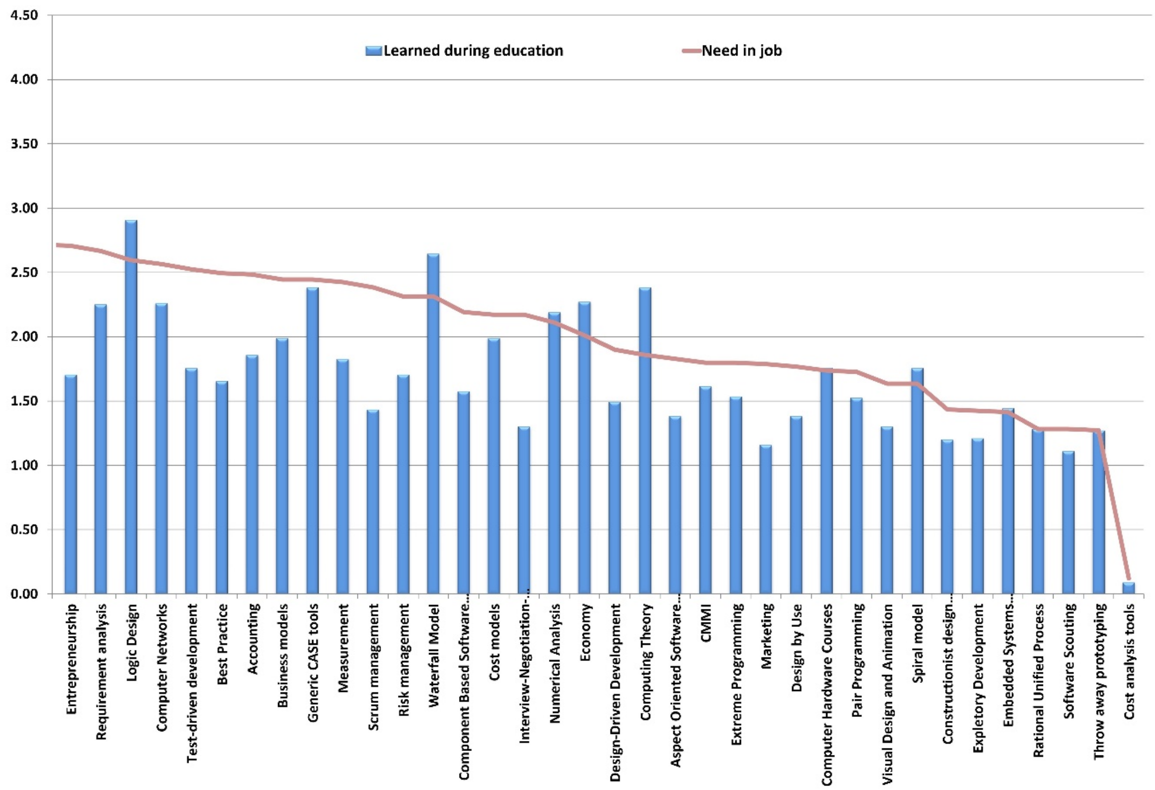

Figure 4. Executive summary of knowledge gap as reported by professionals (Cont.)

Table 6 shows the ten areas in which the survey respondents indicated the greatest knowledge gap. The widest gap occurs in "Communication Skills" and "Customer Relationships. The gaps in "Leadership" and "Team Working", also in the same category as General and Personal Skills, lead to the conclusion that institutions might consider improving soft skills by incorporating social science courses and even enlisting support from corporations. It is noteworthy that the general Computer Science topics show no significant knowledge gap, suggesting that information science curricula have adjusted to address the needs of the software industry in terms of the core computer science knowledge technical competences. However, it seems that these curricula have neglected soft skills (Soft skills fall under the category General and Personal Skills in the survey). As the IT industry undergoes a rapid change and expansion, the depth and breadth of the projects increases, and projects teams may work across continents. Effective communication skills may therefore be needed to address the differences in technical competency of project partners.

Table 6. Knowledge gap (Needed-Learned)

	Subject/Field	Learned	Needed	Δ	Category
1	Communication skills (PS)	2.26	3.75	1.49	Personal Skills
2	Customer relationships (PS)	1.60	3.05	1.45	Personal Skills
3	Test (SPM)	2.26	3.51	1.25	SW Project & Processes
4	Agile Software Development (SDEV)	1.93	3.11	1.18	Development & Coding
5	Leadership (PS)	2.12	3.23	1.11	Personal Skills
6	Team working (PS)	3.01	4.07	1.06	Personal Skills
7	Reuse (SPM)	2.22	3.24	1.02	SW Project & Processes
8	Entrepreneurship (PS)	1.71	2.71	1.00	Personal Skills
9	Implementation (SPM)	2.76	3.76	1.00	SW Project & Processes
10	Test tools (CASE)	1.78	2.77	0.99	CASE Tools &Standards

Among 71 topics, respondents reported that they had more than sufficient knowledge for only 11, with the remaining needing more knowledge/skill. Table 7 shows these 11 topics. Based on this information, some topics can be de-emphasized in the curricula. Waterfall and spiral software development models, which previously dominated software development models, have been replaced by others, such as scrum and agile. The increasing popularity of agile development methodology is also evident from Table 5, which shows it has the fourth largest knowledge gap.

Table 7. Surplus knowledge (Needed - Learned)

	Subject/Field	Learned	Needed	Δ	Category
1	Computing Theory	2.38	1.86	-0.53	General Computer Science
2	Waterfall Model	2.65	2.31	-0.33	Model, Method & Frameworks
3	Logic Design	2.91	2.60	-0.31	General Computer Science
4	UML	3.08	2.78	-0.30	CASE Tools &Standards
5	Economy	2.27	2.01	-0.26	General Skills
6	Spiral model	1.76	1.64	-0.12	Model, Method. & Frameworks
7	Numerical Analysis	2.19	2.11	-0.08	General Computer Science
8	Embedded Systems Prog.	1.44	1.41	-0.03	Development & Coding
9	Computer Hardware Courses	1.76	1.74	-0.02	General Computer Science
10	Rational Unified Process	1.28	1.28	0.00	Model, Method. & Frameworks
11	Throw away prototyping	1.27	1.27	0.00	Model, Method. & Frameworks

A notable observation is that hardware-related courses exhibit a surplus knowledge. It is safe to assume that software is gaining greater momentum than hardware, and currently the time spent on hardware-related courses during education is enough for the industry. Computing Theory and Numerical Analysis – the two topics under general computer science category closest to Mathematics – are considered to be given sufficient attention, More specifically, Computing Theory is in first position, a surprising result, because of its rather indirect connection between with industrial needs and because this is a fundamental computer science course, laying computational foundations. A plausible explanation for its performance is its contribution to wide variety of other technical competencies.

Table 8 shows executive summary of the knowledge gap for the leading two topics in each category and their importance based on Software Professionals' responses. The importance of each topic is defined as the professionals' assessment of demand for this skill. Topics considered important, but undertaught, such as Agile Development, Test and Communication Skills suggests a need for attention.

Table 8. Executive summary of gaps

Topics	Under Learnt	Over Learnt	Importance
General Computer Science	1. Security 2. Database	1. Computing Theory, 2. Logic	• Database Systems, • Data Structure and Algorithms, • Security
Development and Coding	1. Languages for Web Programming, 2. Implementation Patterns	1. Embedded Systems Programming, 2. Visual Design and Animation	• Concepts of Programming Languages, • Object Oriented Programming • More than one Programming Languages,
SW Project Phases, Processes & Management	1. Test 2. Reuse	1. Cost models 2. Requirement gathering	• Implementation • Test • Design
CASE Tools and standards	1. Test tools, 2. Documentation Tools	1. UML 2. Cost analysis tools	• Documentation tools • UML • Test tools
Model, Methodologies & Frameworks	1. Agile 2. Scrum	1. Waterfall Model 2. Spiral model	• Object Oriented Software Engineering, • Agile Software Development, • Iterative and Incremental Development
General Skills	1. Communications Skills 2. Customer Relationships	1. Economy, 2. Ethical issues	• Team working • Analytic thinking • Communication skills

Academics vs Employees

Tables 9 and 10 show the academics' responses. The difference in mean values of the answers given by academics (i.e., taught during education) and employers (i.e., skill expected in workplace) indicates the degree to which the skill/knowledge taught fits with the workplace expectations. A different perspective of the educational knowledge gap is provided by the academics' and employers' responses. The employers were asked *"How would you rate your skill expectations from IT or software professionals about the following subjects/fields?"* (0: N/A; 1: Very Low; 2: Low; 3: Medium; 4: High; 5: Excellent) and the academics were asked *"Which of the following areas/subjects do you teach in your department/program?"* (0: No Idea; 0: We do not teach; 4: We teach; 5: We teach with hands-on example). The educational knowledge gap refers to the gap between the content of formal education and the employer's skill expectation, as perceived by these stakeholders.

Table 9. Academics and employer responses about computer science and software engineering-based courses and skills

Category/Subject/Field	Academics		Employers		p (<0.05)	Meaningful Difference?	Increase/ Decrease in Demand
	Mean	SD	Mean	SD			
General Computer Science							
Computer Networks	4.03	0.67	2.82	1.67	0.0003	Yes	↓
Data Communication	2.85	0.99	2.92	1.85	0.7162	No	↑
Data Structure and Algorithms	3.94	0.70	3.82	1.64	0.9380	No	↓
Logic Design	2.56	1.06	3.18	1.90	0.1013	No	↑
Database Systems	3.99	0.69	4.26	1.25	0.1960	No	↑
Computing Theory	3.10	0.86	2.71	1.86	0.3979	No	↓
Computer Hardware Courses	3.01	0.96	2.11	1.54	0.0363	Yes	↓
Operating Systems	3.65	0.80	2.84	1.69	0.0452	Yes	↓
Security (Web, Network, etc.)	3.13	0.91	3.11	1.75	0.8708	No	↓
Numerical Analysis	2.39	1.06	2.80	1.95	0.2614	No	↑
Development and Coding							
Concepts of Programming Lang.	3.81	0.75	4.26	1.35	0.0785	No	↑
Object Oriented Programming	4.22	0.62	4.21	1.44	0.7197	No	↓
More than one Programming Lang	4.14	0.65	4.08	1.36	0.8627	No	↓
Implementation Patterns	2.92	0.95	3.58	1.77	0.0669	No	↑
Embedded Systems Programming	2.74	1.01	2.45	1.88	0.6184	No	↓
Languages for Web Programming	3.43	0.87	3.79	1.71	0.2344	No	↑
Visual Design and Animation	2.10	1.16	2.61	1.50	0.1700	No	↑
SW Project Phases, Processes & Management							
Requirement gathering	2.99	0.91	3.45	1.91	0.1775	No	↑
Feasibility studies	2.00	1.15	3.32	1.82	0.0009	Yes	↑
Cost models	1.79	1.18	2.66	1.71	0.0198	Yes	↑
Business models	1.60	1.22	3.00	1.79	0.0002	Yes	↑
Requirement analysis	2.97	0.90	3.40	1.88	0.2174	No	↑
Design	3.64	0.78	4.00	1.43	0.1718	No	↑
Architecture	3.51	0.82	3.63	1.75	0.5549	No	↑
Test	3.15	0.88	3.87	1.44	0.0356	Yes	↑
Verification and Validation	3.24	0.86	3.66	1.58	0.1759	No	↑
Quality	2.58	1.01	3.95	1.27	0.0002	Yes	↑
Measurement	2.58	1.01	3.13	1.61	0.1171	No	↑
Maintenance	2.10	1.13	3.55	1.57	0.0002	Yes	↑
Reuse	2.14	1.11	3.24	1.88	0.0055	Yes	↑
Implementation	3.60	0.78	4.00	1.47	0.1612	No	↑
Configuration management	2.07	1.09	3.32	1.76	0.0018	Yes	↑
Risk management	1.82	1.17	2.79	1.73	0.0125	Yes	↑
Project management	3.32	0.84	3.76	1.44	0.1333	No	↑
CASE Tools and standards							
Generic CASE tools	2.13	1.13	2.45	1.97	0.3805	No	↑
UML	3.29	0.87	2.47	1.91	0.0705	No	↓
Documentation tools	2.47	1.07	3.05	1.72	0.1178	No	↑
Test tools	2.39	1.06	2.87	1.91	0.2068	No	↑
CMMI	1.24	1.27	1.76	1.78	0.1370	No	↑
Cost analysis tools	0.93	1.34	1.82	1.83	0.0110	Yes	↑

Table 10. Academics and employer responses about software engineering methodologies and approaches, and general skills

Category/Subject/Field	Academics Mean	Academics SD	Employers Mean	Employers SD	p (<0.05)	Meaningful Difference?	Increase/ Decrease in Demand
Model, Methodologies &Frameworks							
Agile Software Development	2.49	1.06	2.18	2.01	0.5838	No	↓
Aspect Oriented Software Eng.	1.15	1.31	1.68	1.86	0.1314	No	↑
Best Practice	2.01	1.11	2.16	2.18	0.6456	No	↑
Component Based Software Eng.	1.58	1.23	2.18	2.08	0.1187	No	↑
Constructionist design methodology	0.74	1.37	1.34	1.65	0.0592	No	↑
Design by Use	1.06	1.31	2.03	2.01	0.0108	Yes	↑
Design-Driven Development	1.44	1.24	2.16	1.97	0.0638	No	↑
Expletory Development	0.36	1.43	1.00	1.53	0.0150	Yes	↑
Extreme Programming	1.61	1.22	1.37	1.67	0.6080	No	↓
Object Oriented Software Eng.	3.29	0.84	3.29	2.18	0.8428	No	↓
Pair Programming	1.51	1.12	1.24	1.57	0.5470	No	↓
Iterative &Incremental Develop.	1.75	1.11	2.08	2.14	0.3887	No	↑
Rational Unified Process	1.60	1.21	1.18	1.66	0.3504	No	↓
Scrum management	1.54	1.20	1.61	1.94	0.8001	No	↑
Spiral model	1.83	1.15	1.53	1.87	0.5245	No	↓
Software Scouting	0.31	1.43	1.05	1.63	0.0046	Yes	↑
Test-driven development	1.74	1.15	1.92	1.85	0.5779	No	↑
Throw away prototyping	1.03	1.27	1.16	1.60	0.6570	No	↑
Waterfall Model	2.46	1.03	2.11	2.01	0.4830	No	↓
General Skills							
Customer relationships	0.89	1.35	3.42	1.61	0.0000	Yes	↑
Communication skills	3.08	0.93	3.66	1.65	0.0881	No	↑
Interview-Negotiation-Contract management	0.57	1.40	2.29	1.83	0.0000	Yes	↑
Team working	3.72	0.78	4.32	1.40	0.0345	Yes	↑
Marketing	0.68	1.39	1.95	1.63	0.0001	Yes	↑
Entrepreneurship	1.63	1.22	2.95	1.82	0.0006	Yes	↑
Ethical issues	2.61	1.02	4.26	1.39	0.0000	Yes	↑
Leadership	1.60	1.23	3.42	1.57	0.0000	Yes	↑
Presentation skills	3.40	0.86	3.68	1.47	0.3052	No	↑
Economy	1.47	1.24	2.45	1.61	0.0064	Yes	↑
Accounting	0.96	1.34	2.74	1.62	0.0000	Yes	↑
Analytic thinking	2.81	0.99	4.40	1.41	0.0000	Yes	↑

Figure 5 ranks the ten leading topics based on the answers given by Academics and Employers. As in Kitchenbaum et al. (2005), rather than depicting the gap as a numerical value, this figure compares the rank orders, showing the overall importance of the topics. The actual gap information is given in Tables 11 and 12.

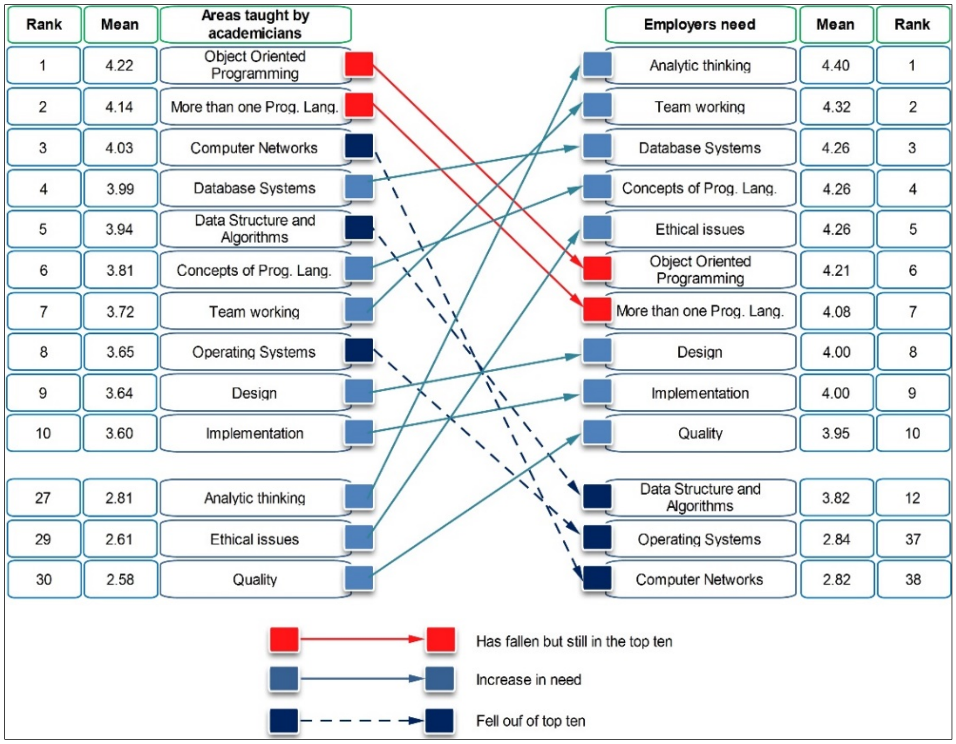

Figure 5. Rank orders of importance of topics

Table 11 reveals some significant results with serious implications for certain courses. Operating Systems and Computer Networks are not considered essential by the employers, but these courses are taught extensively in universities. The two most popular academic topics, Object Oriented Programming and Programming Languages, continue to be desired by employers, but Computer Networks, the third most commonly taught topic, along with Operating System skills, seem much less important for employers.

Table 11. The educational knowledge gap between employer skill expectations and what is taught by the academics based on the mean values.

	Topic	Academics Mean	Employer Mean	Δ Mean	Academics Teaching Rank	Employer Skill Expectation Rank
1	Customer relationships	0.89	3.42	2.53	66	22
2	Leadership	1.60	3.42	1.82	54	23
3	Accounting	0.96	2.74	1.78	64	41
4	Interview-Negotiation, Contract management	0.57	2.29	1.72	69	49
5	Ethical issues	2.61	4.26	1.65	29	5
6	Analytic thinking	2.81	4.40	1.59	27	1
7	Maintenance	2.10	3.55	1.46	41	20
8	Business models	1.60	3.00	1.40	52	33
9	Quality	2.58	3.95	1.36	30	10
10	Entrepreneurship	1.63	2.95	1.32	50	34

Another conclusion drawn from Figure 5 is that Computer Networks, Computer Hardware Courses, UML, Operating Systems and Computing Theory are over-taught, showing the most discrepancy with employer expectations.

A number of further insights arise such as, increasing expectations for software project phases, process and management skills as indicated by Tables 9 and 10. To illustrate, Design, Implementation, and Quality are in the ten most needed topics (ranked 8th, 9th and 10th respectively by the employers), and Test, Project Management, Verification-Validation and Architecture are in the 20 most needed (ranked 11th, 14th, 16th and 17th respectively by the employers). Academia therefore gives adequate emphasis to Software project phases, process and management skills. Additionally, Tables 9 and 10 reveal that the 20 most taught topics include those also considered important by the employers - Design (9th), Implementation (10th), Test (18th), Project Management(14th), Verification-Validation (17th) and Architecture (11th).

UML is a rather different case. For software engineering departments, all academics say UML is an inevitable part of specification, while industry emphasizes agile development more than documentation. UML's position is very low in employer expectations, and the survey shows that it is excessively taught; however, all other CASE tools and standards topics are taught exactly in line with employer needs. An unexpected result is that Software Development Models, Methodologies, & Frameworks are considered less important than other topics, including the agile and scrum software development. This result is particularly interesting, highlighting that agile and scrum development models are gaining momentum. One possible explanation is that software is still being developed using nonstandard and ad-hoc methods.

The results show that General and Personal Skills are rapidly gaining importance. In this category Analytical Thinking and Team Working topics are the two competencies most demanded by IT and software professionals. Other important topics in this area are ethical values, ranked fifth by employers, and Presentation and Communications skills, ranked 15th and 16th respectively, as shown in Tables 9 and 10.

The difference between the mean values of employers and mean values of academics' responses is another measure of the educational knowledge gap. Table 11 shows this analysis, ordering the topics based on the mean differences, and giving the rank orders of the topics taught by academics, and IT skills expected by the employer. Apparently neglected by formal education are soft skills such as Customer relationships, Leadership, and Accounting, and Interview-Negotiation-Contract management.

Table 12 provides a similar analysis, this time based on the rank order differences. The results exhibit almost the same tendencies as in Table 11. Customer relationships category is ranked 66th by the academics, but 22nd by the employers. Figures 6 and 7show an executive summary of asymmetries of subjects between employer expectations and those taught (or included in curricula).

Table 12. The educational knowledge gap between employer skill expectations and what is taught by the academics based on rank order differences.

	Topics	Academics Teaching Rank	Employer Skill Expectation Rank	Δ
1	Customer relationships	66	22	44
2	Leadership	54	23	31
3	Analytic thinking	27	1	26
4	Ethical issues	29	5	24
5	Accounting	64	41	23
6	Maintenance	41	20	21
7	Quality	30	10	20
8	Interview-Negotiation-Contract management	69	49	20
9	Feasibility studies	44	25	19
10	Business models	52	33	19

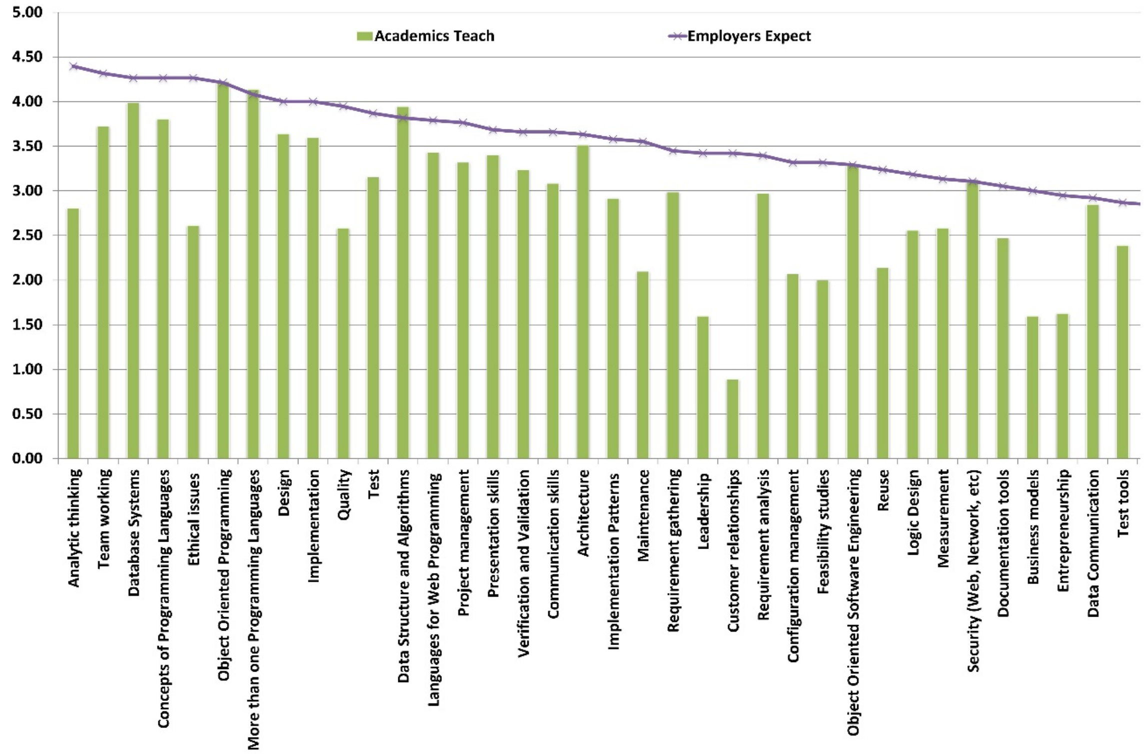

Figure 6. Asymmetry between academics and employers

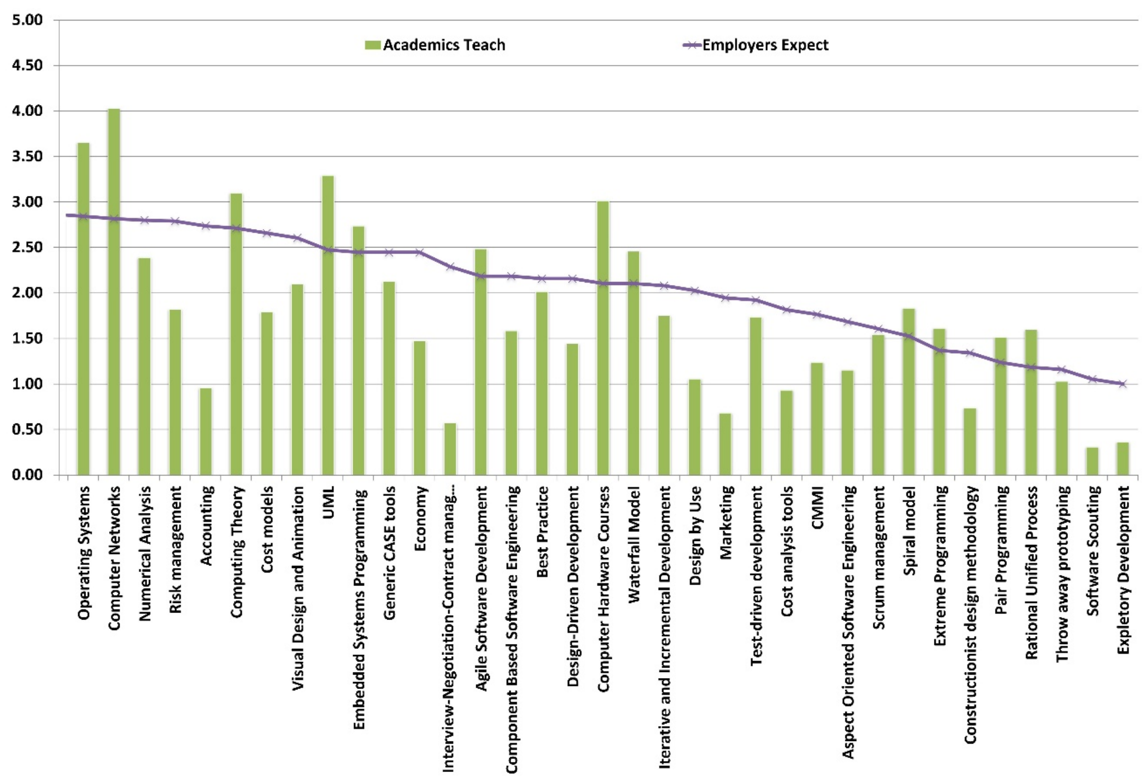

Figure 7. Asymmetry between academics and employers (Cont.)

RECOMMENDATIONS

The survey conducted in this study serves to two major goals: first, investigate mismatches between the existing Information Technology education system and industry needs, and, second, propose possible solutions to reduce these. Upon completion of the survey, we presented early results of the study to a very large conference audience, who provided very significant feedback, including on the differences between education systems (Sahin & Celikkan 2017). The majority of the feedback and comments concerned how to utilize the results to improve existing curriculums in different education systems. (We learned that several universities apply different teaching methods, such as distance education and flipped learning, rather than a conventional teaching.) In further research, we plan to investigate how to improve the skills of students by taking specific teaching environments into consideration. Several conference participants emphasized the importance of program accreditation, explaining how this can improve students' skills and increase the variety of skills. Based on this feedback, we investigated several accreditation institutions, mainly in engineering, in order to determine their level of awareness of the asymmetry between academy and industry, and the solutions they propose (ABET, https://www.abet.org/; ABEEK, http://abeek.knu.ac.kr/Keess/comm/support/main/english.jsp; JABEE, https://jabee.org/en/; MUDEK, http://www.mudek.org.tr/en/ana/ilk.shtm).

The recommendations given below may not suit all types of IT education system but can at least be taken into consideration during the setting up or revision of a program curriculum.

- Accreditation is very important for IT programs for establishing an effective education program and aligning IT programs with the realities of business life. The strict rules and requirements of the accreditation process help institutions to reduce the knowledge and skill gaps, as pointed out in this study. Accreditation is a lengthy process, and unfortunately, most of the accreditation institutions do not accept the program applications until two years after awarding their first degrees to graduates. The accreditation process requires implementing certain improvement cycles, a time-consuming process, therefore, candidate programs should fulfill the accreditation requirements as soon as possible. Therefore, IT programs, especially the more recent ones, should proactively follow the regulations of accreditation institutions, supplemented with views from experienced and reputable higher education institutions for future accreditation processes when determining their curricula and approach to education.
- Establishing multiple learning tracks in IT education from third year would help to equip students with knowledge and skills in specialized IT fields. These tracks should include courses conducted in collaboration with industrial stakeholders and focus on specific industry problems and their solutions. Meanwhile, students should be encouraged to complete their summer work experience with relevant IT industry companies in order to prepare them for the various fields of IT sector, providing experience of the work environment before graduation (Rowlinson et al., 2019).
- General Education Courses (GED) are incredibly significant part of engineering education (Jamieson & Shaw, 2019; Trevelyan 2019). They aim to enhance students' social skills by equipping them with capabilities such as analytic thinking, team working, and ethical perceptions. Although important, GED courses in the curricula should be carefully chosen for their relevance and benefits, and the number of GED courses should not exceed the number of profession courses.
- Technical electives in education are an indispensable part of curricula for improving students' field knowledge, therefore, a large number of well-organized technical electives should be offered. They should be organized in specialized elective pools (this activity should be done in parallel, to create learning tracks) and grouped according to related fields.

- In higher education, advisors play a crucial role in shaping students' future. An important activity is guiding students' course enrollments according to their interests. However, the survey has revealed an interesting result: some academics believe advisors should not be involved in course selection, leaving students free to select their preferred courses. In fact, especially at the beginning of academic life, students should be closely guided, and advisors should recommend alternatives, due to new students' inexperience and lack of knowledge of business life. This does not mean that advisors must force students into predefined tracks, but rather highlight different alternatives to help students to understand their abilities and interests.

SURVEY LIMITATIONS

Although each group (professionals, academics, and employers) had at least 30 participants, due to size differences, comparisons were made with respect to averages of answers. Participants' IT priorities are influenced by their countries' development strategies and cultural aspects, and therefore expectations may vary; however, we interpreted the results as indicating that information technology is an indispensable part of life globally, and it should be high priority in market. In addition, to causing academics any discomfort about their knowledge level, rather than a Likert scale (i.e., 0 to 5), the academics' survey employed a simple yes/no scale, for example, "I teach/I do not teach", and then answers were normalized into numbers.

CONCLUSIONS

A quantitative analysis of the knowledge difference between software industry expectations and the skills taught in higher education is crucial in the design and update of higher education curricula. Such an analysis allows greater cooperation with the software industry to meet current computing needs. For this purpose, we conducted a survey with questions in 6 categories and on 71 topics with three target groups: employers, software professionals, and academics. The results indicate a lack of emphasis on personal and non-technical skills in undergraduate education, in comparison with general computer science, development and coding courses. A noticeable lack of knowledge in non-technical skills is observed when compared to technical skills, consistent with previous studies. Employers' and software experts' responses emphasize that soft skills, in particular analytical thinking and teamwork, should not be ignored. The survey shows that only 15% of the selected subjects were taught sufficiently or more than sufficiently. Computing theory, numerical analysis, hardware, logic courses, spiral and waterfall development methodologies are among over-taught subjects requiring reconsideration. As IT sector undergoes a rapid transformation in reaction to fast-paced developments and innovations, spiral and waterfall models should be replaced in the curricula by emerging software development models such as Agile; the survey shows that Agile Software Development has the fourth largest information gap. A multidisciplinary approach to teaching soft skills should include areas such as communication, ethics, leadership, and customer relations, instead of relying solely on IT based disciplines.

The software professional and employers provided invaluable feedback in the free form comments, summarized below.

- Rather than a theoretical emphasis, courses should include hands-on projects that address the practical industry needs. Innovative project topics should be designed in collaboration with industry.
- The number of courses should be reduced to ensure a practical curriculum focused on industry needs, especially in the final two years of the program courses.
- An effective communication channel should be established between students and industry.
- It is important to reduce the distance between academics and students and provide an interactive environment for technical discussions on software engineering.

- Undergraduates should be exposed to newer technologies and methodologies, such as scrum development, NoSQL, and configuration management tools TFS, Git, SVN and test tools such as Junit, jasmine, and PHPunit. Development teams are often dispersed across continents, and scrum development is a viable methodology in addressing issues stemming from distributed development.
- Advantage in the job market are provided by Enterprise level computing and development knowledge, such as JEE, ASP.net, and Framework use. Instead of focusing on a programming language, it is more beneficial to gain a deeper understanding of the ecosystem that surrounds the language, such as API's, protocols, and tools, enabling the development of applications using that language.
- Current development methodologies and tools reduce the importance of documentation in higher education.

The feedback provided by employers focused more on general, softer skills rather than technical skills.

- Software engineering process standards should be understood and followed.
- Project Management and teamwork topics should have a greater emphasis in the curriculum.
- Employees should demonstrate motivation, ambition, and ethical sensitivity.
- Students should increase knowledge of methodology rather than programming languages.
- Distinguish between future software developers and future academics. Admit the first group into yearlong internship and co-op programs.
- It is difficult for companies to retain newly recruited graduates after training and educating them because they use their knowledge and skill to find work elsewhere.

REFERENCES

Aasheim, C. L., Li, L., & Williams, S. (2009). Knowledge and skill requirements for entry-level information technology workers: A comparison of industry and academia. *Journal of Information Systems Education*, *20*(3), 349-356. https://aisel.aisnet.org/jise/vol20/iss3/10

Anderson, J., & Rainie, L. (2018). *The future of well-being in a tech-saturated world.* Pew Research Center. http://www.pewinternet.org/2018/04/17/the-future-of-well-being-in-a-tech-saturated-world

Arellano, A., & Cámara, N. (2017). *The importance of ICT in society's needs: An empirical approach through Maslow's lens.* BBVA Research. https://www.bbvaresearch.com/en/publicaciones/the-importance-of-ict-in-societys-needs-an-empirical-approach-though-maslows-lens/

Beckman, K., Coulter, N., Khajenoori, S., & Mead, N. R. (1997). Collaborations: closing the industry–academia gap. *IEEE Software*, *14*(6), 49-57. https://doi.org/10.1109/52.636668

Benamati, J. H., Ozdemir, Z. D., & Smith, H. J. (2010). Aligning undergraduate IS curricula with industry needs. *Communications of the ACM*, *53*(3), 152–156. https://doi.org/10.1145/1666420.1666458

Brynjolfsson, E., & Saunders, A. (2010). *Wired for innovation: How information technology is reshaping the economy*. The MIT Press. https://doi.org/10.7551/mitpress/8484.001.0001

Burnell, L. J., Priest, J. W., & Durrett, J. R. (2002). Teaching distributed multidisciplinary software development. *IEEE Software*, *19*(5), 86-93. https://doi.org/10.1109/MS.2002.1032859

Callahan, D., & Pedigo, B. (2002). Educating experienced IT professionals by addressing industry's needs. *IEEE Software*, *19*(5), 57-62. https://doi.org/10.1109/MS.2002.1032855

Cappel, J. J. (2002). Entry-level IS job skills: A survey of employers. *Journal of Computer Information Systems*, *42*(2), 76-82. https://www.tandfonline.com/doi/abs/10.1080/08874417.2002.11647490

Carrington, D., Strooper, P., Newby, S., & Stevenson, T. (2005). An industry/university collaboration to upgrade software engineering knowledge and skills in industry. *Journal of Systems and Software*, *75*(1–2), 29–39. https://doi.org/10.1016/j.jss.2004.02.020

Celikkan, U., & Sahin, Y. G. (2017). Perceived information gap between academia and industry: A qualitative study. *In Proceedings of 9th Annual International Conference on Education and New Learning Technologies*, Barcelona (Spain). 3rd - 5th of July 2017, 9756-9765. https://doi.org/10.21125/edulearn.2017.0849

Dennis, J. B. (1969). Inter-relating hardware and software in computer science education. *In Proceedings AFIPS 1969 SJCC.* AFIPS Press, Montvale, 537-538. https://doi.org/10.1145/1476793.1476874

de Winter, J., & Dodou, D. (2010). Five-point Likert items: t Test versus Mann–Whitney–Wilcoxon. *Practical Assessment, Research and Evaluation, 15*(11), 1-16. https://www.semanticscholar.org/paper/Five-Point-Likert-Items%3A-t-Test-versus-Winter-Dodou/68f3c1623325aafefd00b4d2a5c1fc841a7ca6e7

Föll, P., Hauser, M., & Thiesse, F. (2018). Identifying the skills expected of IS graduates by industry: A text mining approach. *In Proceedings of 39th International Conference on Information Systems (ICIS 2018),* San Francisco, California (USA). Vol 4. 2730-2747. https://aisel.aisnet.org/icis2018/education/Presentations/5/

Gonzalez-Torres, A. (2018). A survey on the tools used in software development and their mapping to the curriculum of computer degrees. *In Proceedings 12th International Technology, Education and Development Conference (INTED)*, 7420-7429. https://doi.org/10.21125/inted.2018.1746

Hayes, J. H. (2002). Energizing software engineering education through real-world projects as experimental studies. *In Software Engineering Education and Training, (CSEE&T), 2002 IEEE 15th Conference on*, 192-206. https://doi.org/10.1109/CSEE.2002.995211

Jamieson, M. V., & Shaw, J. M. (2019). Teaching engineering for a changing landscape. *Canadian Journal of Chemical Engineering*, *97*(11), 2870-2875. https://doi.org/10.1002/cjce.23626

Kim, Y., Hsu, J., & Stern, M. (2006). An update on the IS/IT skills gap. *Journal of Information Systems Education*, *17*(4), 395–402. https://aisel.aisnet.org/jise/vol17/iss4/5

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

Lee, D. M. S., Trauth, E. M., & Farwell, D. (1995). Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation. *MIS Quarterly*, *19*(3), 313-340. https://doi.org/10.2307/249598

Lethbridge, T. (1998). A survey of the relevance of computer science and software engineering education. In *Proceedings 11th International Conference on Software Engineering*, 56-66. IEEE Computer Society Press, Silver Spring. https://doi.org/10.1109/CSEE.1998.658300

Lethbridge, T. (2000). What knowledge is important to a software professional? *Computer*, *33*(5), 44-50. https://doi.org/10.1109/2.841783

Mead, N. R. (2009). Software engineering education: How far we've come and how far we have to go. *Journal of Systems and Software*, *82*(4), 571–575. https://doi.org/10.1016/j.jss.2008.12.038

Medlin B. D., Dave, S. D., & Vannoy, S. A. (2001). Students' views of the importance of technical and non-technical skills for successful IT professionals. *Journal of Computer Information Systems*, *42*(1), 65-69. https://www.tandfonline.com/doi/abs/10.1080/08874417.2001.11647040

Merhout, J. W., Havelka, D., & Hick, S. N. (2009). Soft skills versus technical skills: Finding the right balance for an IS curriculum. *In Proceedings of 15th Americas Conference on Information Systems* (AMCIS 2009), (paper 9). https://aisel.aisnet.org/amcis2009/9

Moreno, A. M., Sanchez-Segura, M. I, Medina-Domingue, F., & Carvajal, L. (2012). Balancing software engineering education and industrial needs. *Journal of Systems and Software*, *85*(7), 1607-1620. https://doi.org/10.1016/j.jss.2012.01.060

Penzenstadler, B., Mahaux, M., & Heymans, P. (2013). University meets industry: Calling in real stakeholders. *In Software Engineering Education and Training (CSEE&T), 2013 IEEE 26th Conference on*, 1-10. https://doi.org/10.1109/CSEET.2013.6595231

Pilgrim, C. J. (2013). Industry involvement in ICT curriculum: A comparative survey. *In Proceedings of 2013 35th International Conference on Software Engineering (ICSE 2013)*, 1148-1153. San Francisco, CA. https://doi.org/10.1109/ICSE.2013.6606666

Plaza, I., Igual, R., Medrano, C., & Rubio, M. Á. (2013). From companies to universities: Application of R&D&I Concepts in higher education teaching. *IEEE Transactions on Education*, *56*(3), 308-315. https://doi.org/10.1109/TE.2012.2218247

Plice, R. K., & Reinig, B. A. (2007). Aligning the information systems curriculum with the needs of industry and graduates. *Journal of Computer Information Systems*, *48*(1), 22-30. https://www.tandfonline.com/doi/abs/10.1080/08874417.2007.11645992

Preston, J. A. (2004). Utilizing authentic, real-world projects in information technology education. *ACM SIG-ITE Newsletter*, *1*(2), 4.1-4.10. https://doi.org/10.1145/1869667.1869671

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

Rowlinson, S. C., Burg, T. C., Bridges, W. C., & Burg, K. J. L. (2019), Enhancing the academic innovation culture by incorporation of customer-centric practices. *Technology and Innovation, 21*(1), 63-74. https://doi.org/10.21300/21.1.2019.63

Sahin, Y. G., & Celikkan, U. (2017). What industry needs and what we teach in information technology: A quantitative study. *In Proceedings of 9th Annual International Conference on Education and New Learning Technologies,* Barcelona (Spain). 3rd - 5th of July 2017, 9775-9785. https://doi.org/10.21125/edulearn.2017.0851

Skevoulis, S. (2011). Engineering a successful partnership between academia and the financial industry: A software engineering program for IT professionals. *In Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on*, 343 – 350. https://doi.org/10.1109/CSEET.2011.5876105

Tesch, D. B., Braun, G. F., & Crable, E. A. (2008). An examination of employers' perceptions and expectations of is entry-level personal and interpersonal skills. *Information Systems Education Journal*, *6*(1), 3–16. http://isedj.org/6/1/

Trevelyan, J. (2019). Transitioning to engineering practice. *European Journal of Engineering Education*, *44*(6), 821-837. https://doi.org/10.1080/03043797.2019.1681631

Wang, C-S., & Li, C-C. (2000). An assessment framework for information technology integrated instruction. *In Proceedings International Conference on Computers in Education/ International Conference on Computer-Assisted Instruction (ICCE/ICCSI) 2000 Teaching and Learning Processes* (Taipei-Taiwan, Nov 21-24), pp.30-37.

Wasserman, A. I., & Freeman, P. (1978). Software engineering education: status and prospects. *Proceedings of the IEEE*, 66(8), pp. 886-892. https://doi.org/10.1109/PROC.1978.11050

ACKNOWLEDGEMENTS

We sincerely thank Simon Mumford for proofreading the article.

BIOGRAPHIES

Yasar Guneri Sahin is a professor in the department of software engineering at Izmir University of Economics, where he has been since 2007. From 2011 to 2014, he served as Department Chair. He has been serving as Dean of Engineering since 2014. During 2018-2019 he was a visiting professor at the University of Missouri Columbia. He received BSc degree from Middle East Technical University in 1994, and MSc and PhD degrees from the Gazi University in 1996 and 2005, respectively. His research interests span real-time systems, software engineering and instructional technologies.

Ufuk Celikkan received his MSc and PhD degrees from North Carolina State University, Raleigh, USA, and BSc degree from Middle East Technical University, Ankara, Turkey. He worked as an Advisory Software Engineer in the Server and Software Divisions of IBM, Austin, Texas and then in the Smart Card Division of Gemalto Inc. also in Austin, Texas. He is currently an Assistant Professor in the Department of Software Engineer at Izmir University of Economics.