AAEE2017 CONFERENCE
Manly, Sydney, Australia

Attitudes Towards Software Engineering Education in the New Zealand Industry

Catherine Watson and Kelly Blincoe
Department of Electrical and Computer Engineering
University of Auckland, New Zealand
Corresponding Author Email: c.watson@auckland.ac.nz

SESSION C2: Interdisciplinary and cross-disciplinary engineering programs and learning environments

CONTEXT Software Engineering (SE) is one of the newest engineering disciplines, and the ideal structure for an SE programme is not yet well understood. The purpose of this study was to obtain feedback on topics that should be covered in an SE degree from Alumni of our SE degree and local professionals working in the SE industry involved in hiring decisions.

PURPOSE We wanted to identify what we are doing well in our current programme, what is valued by industry, and what new areas to include or improve. Our findings can be used by other SE programmes as they update their cirriculum. We also wished to investigate whether there was a perceived value in SE being a four-year engineering degree and what the perceived differences were with a three-year Computer Science (CS) degree.

APPROACH We created two online surveys to get feedback on our SE programme, one for SE Alumni and one for local SE professionals. The surveys, included Likert scale and open-ended questions. Respondents rated the importance of multiple SE-related skills and the ability of local SE professionals in those skills. Respondents also provided their opinions on various aspects of our SE degree.

RESULTS We got responses from 28 Alumni and 22 professionals from the local SE industry. The results provide a ranked list of 37 skills deemed important for SE graduates, which can be used to identify gaps in existing SE programmes. The skills ranked most important were “Working in a team”, “Communication skills” and “Solving problems in a team”. We also compared the differences between the importance of a given skill and the actual ability of SE professionals for that skill as rated by SE managers. The largest gaps were software quality, software testing, software design, communication skills and solving problems in a team.

We also found that the majority of respondents (20 of 26 or 76.9% of Alumni and 11 of 15 or 73.3% of industry respondents) believe there is a difference between the graduates of an SE program compared to a CS program. Most respondents believe that there are unique benefits for each degree and that the degrees complement each other.

CONCLUSIONS Soft skills are critical skills in SE and make up seven of the top eight most important skills. Two soft skills, communication skills and solving problems in teams, were in the top five gaps in current SE programs identified by SE hiring managers. Other areas where SE programs could improve include strengthening the software quality, design and testing skills of graduates. We also found that industry perceives a difference between a SE and a CS graduate, and that both are needed in the software industry.

KEYWORDS Software Engineering, Software Industry, Curriculum

Introduction

Software Engineering (SE) is one of the newest engineering disciplines. For many years, SE was taught as one or two papers in a Computer Science (CS) degree, but recently emerged as an Engineering program in its own right. Previous research found a large mismatch between SE education and industry needs (Leibenberg et al. 2015), and despite initiatives to define a core body of knowledge for SE there remains confusion about what distinguishes SE and CS programmes (see, for example, Landwehr et al. 2017). Thus, the ideal structure for an SE programme is still not yet well understood. SE was introduced at the University of Auckland (UoA) in 2000, with the first graduates in 2003. It has successfully undergone three accreditation processes, the last being in 2015. However, there has been no formal review on how the Software Industry views the degree. The purpose of this study was to obtain feedback on topics that should be covered in an SE degree from UoA SE Alumni and local professionals working in the SE industry involved in hiring decisions for SE-related positions.

The SE program at the UoA is about to undergo a major curriculum review. We wanted to identify what we are doing well in our current programme, what is valued by industry, and what new areas to include or improve. We also wished to investigate whether there was a perceived value in SE being a four-year engineering degree and what the perceived differences were with the three-year CS degree, also offered at the UoA.

Related Work

The term "Software Engineering" has been around since the late 1960s (Landwehr et al. 2017, Parnas 1999), yet its precise definition remains unclear. Parnas (2011) calls SE a discipline that is missing in action. The term was used as a euphemism for "programmer" in job advertisements (Parnas 1999), and the difference between these terms is often unclear even today (Landwehr et al. 2017). The term is also used interchangeably with Software developer (Tookey 2015). Tookey (2015) argues this is to the detriment of the software development profession, because hiring managers focus more on skills related to learnable technologies, rather than focusing on broader SE skills; this is ultimately having an impact on the quality of software products produced.

The ambiguity in the definition also impacts on how SE is taught. There have been a number of efforts to define a core body of knowledge for SE. The *Software Engineering Body of Knowledge* (SWEBOK)[1] is an IEEE Computer Society publication which identifies 15 knowledge areas: requirements, design, construction, testing, maintenance, configuration management, engineering management, engineering process, methods, quality, professional practise, engineering economics, computer foundations, mathematical foundations, and engineering foundations. It was first published in 2004 and is now up to its third version. SWEBOK was intended as a guide to what skills a SE graduate should have after 4 years of training. However, it covers only material that is specific to SE. The *Curriculum guideline for Undergraduate Degree Programs in Software Engineering* (SE2014)[2] is another IEEE Computer Society publication, which is similar to SWEBOK, but it is specifically aimed at undergraduate program development and also includes material not specific to SE that is necessary for professional engineering training, including mathematical and engineering fundamentals and professional practice skills. Its first version was also published in 2004, and a revised version came out in early 2015. SE2014 identifies 10 knowledge areas: Computing essentials, maths and engineering foundations, professional practise, software modelling and analysis, requirements analysis and specification, software verification and

[1] https://www.computer.org/web/swebok
[2] http://www.securriculum.org/SE2014FinalVersion.pdf

validation, software process, software quality, and security (Ardis 2015). There is a large amount of consensus in the skills required for SE in the two documents, although the knowledge areas are organised differently (Tookey 2015).

There is also literature available on curricula for specific SE programmes using the above principles (e.g. Frezza et al. 2006, Karaunasekera and Bedse 2006, Ramakrishnan 2007, Alarifi et al. 2014, Camilloni et al. 2015). However, Landwehr et al. (2017) argue that instead of looking at curricula and knowledge areas it is better to look at capabilities required for a software engineer to be able to develop products. They note that there is a difference between how science and engineering programs present the bodies of knowledge in SE. The former teaches students how to verify and extend that knowledge; the latter teaches students how to apply that knowledge when developing products. They also argue that the associated body of knowledge will grow quickly in SE, but the capabilities required for software engineers are the fundamentals and will be slow to change. Yet, Moreno et al. (2012) argue there are still gaps between what industry wants and the skill set of the graduates.

The original motivation for *Curriculum guideline for Undergraduate Degree Programs in Software Engineering* SE2004 (SE2014's predecessor) was a survey of 200 participants in the US software industry (Lethbridge 2000). Participants were asked for feedback on 75 education subject areas. The topics were determined by studying typical university curricula in computer science and computer engineering (Lethbridge 2000). The results suggested the need for a new curriculum, which ultimately gave rise to SE2004. To investigate the universality of the industry needs, industry in other countries have also been assessed using the same set of criteria, such as Britain (Kitchenham et al. 2005), Finland (Surakka 2007) and South Africa (Liebenberg et al. 2015). However, these studies used out-dated subject criteria (based on Lethbridge (2000)) and focused only on the technical skills (e.g. databases, programming) and did not include soft skills. Aasheim et al. (2009, 2012) surveyed the US industry with their own set of criteria that included both soft skills and technical skills. In both of their studies, the top 10 skills were the same: honesty and integrity, attitude, willingness to learn new skills, communication skills (written and oral), analytic skills, professionalism, ability to work in teams, flexibility, motivation, and interpersonal skills. However, these surveys focus only on the US industry.

Stevens and Norman (2016) surveyed the Information technology sector about graduate skills, and found that there was an identified need for more soft skills. However, whilst this sector includes Software Engineers, it involves other software professionals, and our specific focus is SE graduates, particularly those from our University, which has the longest running SE program in the country, with the first intake in 2000. The original syllabus was motivated by SWEBOK and later SE2004. The SE profession has since matured, and there is now a better understanding of what SE entails. We were interested in gaining insights on the local industry attitudes towards our curricula and identifying any gaps.

Methodology

Description of Survey

We created two online surveys to get feedback on the SE Program, one for the UoA SE Alumni and one for local SE professionals. The surveys, which had ethics approval, included Likert scale and open-ended questions. Respondents rated the importance of multiple SE related skills (listed in Table 2) and the ability of SE graduates in these skills. Similar to Lethbridge (2000) and Aishem et al. (2009, 2012) the skills were topic areas within SE (based on topics covered by the current UoA curriculum). In addition, we also included key graduate attributes inspired by graduate competencies from the Institute of Professional Engineers New Zealand. To rate the importance of SE related skills, we used an unbalanced (skewed towards the positive) rating scale since we expected mostly positive answers and

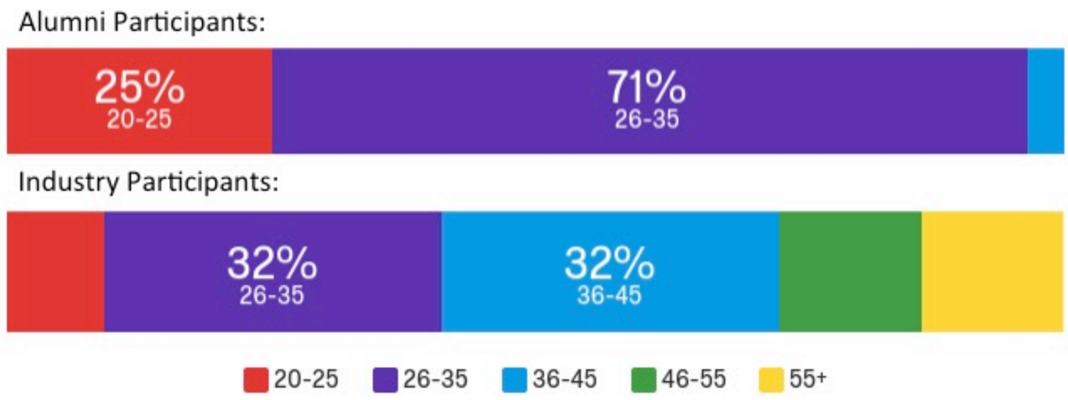

Figure 1: Age of survey participants

wanted to measure the degree of the responses (Parasuraman 2006). Respondents also provided opinions on various aspects of the SE degree. The full survey can be found online [3].

Survey Participants

The survey was sent to UoA SE alumni and professionals working at local software engineering companies. We got responses from 28 Alumni, 26 of whom stated they were currently working in the Software Industry. For the SE professionals, we obtained 22 responses, including 18 who said they are often involved in hiring decisions. We asked each participant several demographic questions including age, gender, years of experience, and education level.

Not surprisingly, our Alumni respondents tend to be relatively young, due to the short duration of the SE program; only one of the Alumni respondents was older than 35 years old. The SE Industry respondents are more varied with respect to age as shown in Figure 1. The Industry participants also had more experience working in the SE industry than the Alumni participants. The Alumni participants had 6.5 years of experience on average, compared to 14 years for the Industry participants. The education level varied across both sets of participants; however, the Alumni participants have a slightly higher education level on average. This is likely due to the SE degree being a joint Bachelor/Honours degree, provided a minimum grade point average is obtained. The highest level of education for each set is shown in Table 1. The gender of our participants is in line with the current ratios in industry, with 16% of our respondents identifying as female and the remainder identifying as male.

Table 1: Highest education level of participants

Education Level	Alumni	Industry
High school degree	0%	13%
Bachelor degree	32%	35%
Honours degree	42%	30%
Masters degree	21%	22%
Doctoral degree	4%	0%

[3] http://kblincoe.github.io/survey/AAEE2017_Alumni_Survey.pdf and http://kblincoe.github.io/survey/AAEE2017_Industry_Survey.pdf

Results

Skills in Software Engineering

In our survey, we asked respondents to rate, using a five-point Likert-type scale, the importance of 37 different skills or knowledge areas. Alumni respondents rated the importance based on the skills needed in their current role, and industry respondents rated the importance based on their hiring decisions for SE related positions. The soft skills dominate the top most important skills, with "Working in a team", "Communication Skills" and "Solving problems in a team" being the top three ranked skills. Table 2 shows the average rating for each skill across all respondents. There was not a significant difference between the two groups of respondents.

Table 2: Mean importance of skills across all respondents

Rank	Skill	Rating	Rank	Skill	Rating
1	Working in a team	4.49	19	Software dev. methodologies	3.45
2	Communication skills	4.49	20	Parallel & distributed computing	3.28
3	Solving problems in a team	4.44	21	Requirements engineering	3.26
4	Professionalism	4.34	22	Data structures & algorithms	3.12
5	Software quality	4.19	23	Human computer interaction	3.12
6	Solving problems independently	4.12	24	Computer Networks	3.02
7	Working independently	4.09	25	High performance computing	2.93
8	Ethics	4.07	26	Algorithms for optimisation	2.81
9	Industry experience[4]	4.06	27	Operating systems	2.72
10	Programming	4.02	28	Artificial intelligence	2.40
11	Software design	4.02	29	Machine learning	2.35
12	Software architecture	3.91	30	Formal specification & design	2.33
13	Software testing	3.84	31	Mathematical modelling	2.28
14	Agile and lean software dev.	3.77	32	Digital systems design	2.19
15	Object orientated software dev	3.58	33	Robotics & intelligent systems	2.02
16	Database systems	3.55	34	Embedded systems	2.00
17	Computer security	3.51	35	Microcomputers	1.98
18	Project management	3.49	36	Computer graphics	1.93

[4] Industry experience results based on Industry participants only, as it was not included in the Alumni survey

Table 3: Top five skill gaps identified by Hiring Manager respondents.

Skill or Knowledge Area	Importance	Ability	Difference
Software quality	4.56	2.53	-2.03
Software testing	4.19	2.73	-1.45
Software design	4.19	3.07	-1.12
Communication skills	4.44	3.33	-1.10
Solving problems in a team	4.50	3.47	-1.03

We also asked the hiring manager respondents to rate their satisfaction with the abilities of software engineers they currently employ or manage for the same set of skills. When looking at the differences between the ratings for the importance of a given skill and the ratings for the actual ability of software engineers for that skill, the largest gap identified by the hiring managers was Software Quality. Table 3 shows the mean ratings for importance and ability and the difference between these ratings for the top five largest gaps.

When comparing the SE and CS graduates, the majority of respondents (20 of 26 or 76.9% of Alumni and 11 of 15 or 73.3% of industry respondents) believe there is a difference between the graduates of the two degrees. In addition, most respondents believe that there are unique benefits for each degree and that the degrees complement each other.

Word clouds from Industry participant responses describing the benefits of each degree are shown in Figures 2 and 3. The word clouds were generated by Qualtrics with the degree names removed as stop words. As can be seen, a CS degree was perceived to have the benefit of providing exposure to the latest technology and tools and a deep understanding of technology. The SE degree was described as providing more real-world, practical knowledge with a solid engineering background. In regards to the degree duration, the shorter duration of the CS degree was cited as a financial benefit for students, while the extra year of study was cited as a benefit of the SE degree for the ability to gain more knowledge and experience before entering the workforce.

The majority of SE professionals involved in hiring decisions stated that individual strengths and weaknesses of a candidate would be more important in their hiring decision than the difference between these degrees.

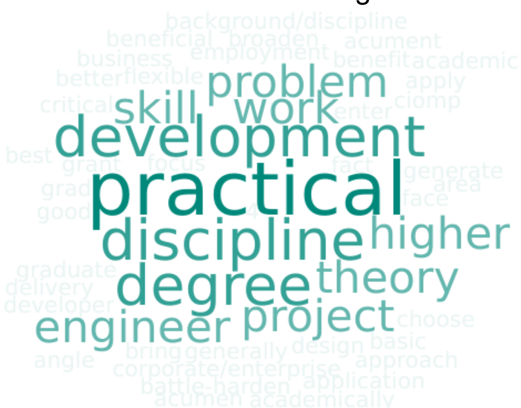

Figure 2: Benefits of Software Engineering

Figure 3: Benefits of Computer Science

Discussion

In terms of importance, our participants rate soft skills very highly. Soft skills dominate the top ten ranked skills, similar to what Aasheim and colleagues (2009, 2012) found. Aasheim et al. (2009) found that US industry ranked interpersonal skills as the most important followed by personal skills and, after that, technical skills. Leibenberg et al. (2015) only asked industry to rate technical skills and found that design skills were the most important skills in the South African Software industry, followed by testing and maintenance. These skills also rank highly in our study. This shows that whilst there will always be local influences on the industry needs, there is commonality across the international community.

We also identified some skill gaps. In the early 2000s, the skill gaps identified by industry were in human computer interfaces, real-time system design, software metrics, software reliability, and requirement gathering (Leibenberg et al. 2000). These gaps had largely disappeared post 2010 (Leibenberg et al. 2015). In our study, the three skills with the greatest gap were software quality, software testing and software design. These have some overlap with the largest skills gap identified by industry in Leibenberg et al. (2015) and in Lethbridge (2000). When comparing the results of our study and those of Lethbridge's 2000 study, the size of the skill gap in both software testing and software design is less in our study. Lethbridge (2000) identified gaps of 1.9 for Testing and 1.7 for Design, compared to our data, which shows gaps of 1.45 and 1.12, respectively. Both studies used a five-point scale, but further investigation is needed to verify if there is a statistically significant difference. The more recent study's findings were similar to ours (Leibenberg et al. 2015). This suggests a potential improvement in the teaching of those subjects over time.

Two soft skills - communication skills and solving problems in a team - were in the top five skill gaps identified by industry. While previous studies found a need for more soft skills (e.g. Moreno et al. 2012, Aasheim et al. 2012), these previous studies had an explicit IT perspective and were not specific to SE. Thus, this is the first SE focused study to identify a soft skills gap. This is interesting; because Tockey (2015) argued that SE hiring managers are not making soft skills explicit in software engineering related jobs and instead focus on more learnable, technical skills. Tockey (2015) argues this is to the detriment of the software industry. Our results indicate that there is at least some awareness by hiring managers of the need to further emphasize and prioritize soft skills.

Limitations

Our survey respondents were self-selected, and their opinions may not generalize to all SE professionals. However, our sample size was comparable to many similar studies described in the related work (eg. Kitchenham et al. 2005, Surakka 2007, Stevens and Norman 2016). Importantly, we obtained responses from Alumni of the SE degree at UoA (of which there is a small pool) and from professionals working at nearly all of the local software engineering companies. Of course, it is likely that many of the Industry respondents do not have SE degrees, so they may not be as familiar with the SE degree as our alumni respondents. However, since the goal of the study was to understand the perceptions of the software engineering degree of those currently employed in the SE industry, the degree our industry respondents obtained is not important. All industry respondents work at software engineering companies and have, on average, more than 13 years of experience in the SE industry.

Conclusion

We conclude that soft skills (like working in a team and communication) are critical skills needed in SE. Thus, SE degrees must include these important components. There are areas where programs could improve, particularly around improving the software quality skills of graduates. Another important finding is that industry does perceive a difference between a SE and a CS graduate, and that both are needed in the software industry.

References

Aasheim, Cheryl L., Lixin Li, and Susan Williams. "Knowledge and skill requirements for entry-level information technology workers: A comparison of industry and academia." *Journal of Information Systems Education* 20.3 (2009): 349.

Aasheim, C., Shropshire, J., Li, L., & Kadlec, C. (2012). Knowledge and skill requirements for entry-level IT workers: A longitudinal study. *Journal of Information Systems Education*, *23*(2), 193.

Alarifi, A., Zarour, M., Alomar, N., Alshaikh, Z., & Alsaleh, M. (2016). SECDEP: Software engineering curricula development and evaluation process using SWEBOK. *Information and Software Technology*, *74*, 114-126.

Ardis, M., Budgen, D., Hislop, G. W., Offutt, J., Sebern, M., & Visser, W. (2015). SE 2014: curriculum guidelines for undergraduate degree programs in software engineering. *Computer*, *48*(11), 106-109.

Camilloni, L., Vallespir, D., & Ardis, M. (2015, May). Using GSwE2009 for the Evaluation of a Master Degree in Software Engineering in the Universidad de la Republica. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2* (pp. 323-332). IEEE Press.

Frezza, S., Tang, M. H., & Brinkman, B. J. (2006). Creating an accreditable software engineering bachelor's program. *IEEE software*, *23*(6).

Karunasekera, S., & Bedse, K. (2007, July). Preparing software engineering graduates for an industry career. In *Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on* (pp. 97-106). IEEE.

Kitchenham, B., Budgen, D., Brereton, P., & Woodall, P. (2005). An investigation of software engineering curricula. *Journal of Systems and Software*, 74(3), 325-335.

Landwehr, C., Ludewig, J., Meersman, R., Parnas, D. L., Shoval, P., Wand, Y., Weiss, D. & Weyuker, E. (2017). Software Systems Engineering programmes a capability approach. *Journal of Systems and Software*, *125*, 354-364.

Lethbridge, T. (2000), ‘Priorities for the education and training of software developers’ Journal of Systekms and Software 53(1), 53-71.

Liebenberg, J., Huisman, M., & Mentz, E. (2015). Software: university courses versus workplace practice. *Industry and Higher Education*, *29*(3), 221-235.

Liebenberg, J., Huisman, M., & Mentz, E. (2015). The relevance of software development education for students. *IEEE Transactions on Education*, *58*(4), 242-248.

Moreno, A. M., Sanchez-Segura, M. I., Medina-Dominguez, F., & Carvajal, L. (2012). Balancing software engineering education and industrial needs. *Journal of systems and software*, *85*(7), 1607-1620.

Parasuraman, A., Grewal, D., & Krishnan, R. (2006). *Marketing research*. Cengage Learning.

Parnas, D. L. (1999). Software engineering programs are not computer science programs. *IEEE software*, *16*(6), 19-30.

Parnas, D. (2011). Software engineering-missing in action: A personal perspective. *Computer*, *44*(10), 54-58.

Ramakrishnan, S. (2007). Accreditation of Monash University Software Engineering (MUSE) Program. *Issues in Informing Science & Information Technology*, *4*.

Stevens, M., & Norman, R. (2016, February). Industry expectations of soft skills in IT graduates: a regional survey. In *Proceedings of the Australasian Computer Science Week Multiconference* (p. 13). ACM.

Tockey, S. (2015). Insanity, Hiring, and the Software Industry. *Computer*, *48*(11), 96-101.

Acknowledgements

We gratefully thank the survey participants for their time and input. We also thank Associate Professor Gerard Rowe and Dr. Keri Moyle for their feedback and encouragement on the initial concept of this paper.