Proposal: Unaligned Expectations in Universities and Industry for New Graduates in Computer Science and Software Engineering

Alan Franzoni
for Submission
Trieste, Italy
alan.franzoni@gatech.edu

Hasti Ghabel
for Submission
Atlanta, GA
hghabel1@gatech.edu

Introduction

There is a widespread agreement that new graduates from computer science and software engineering **do not always possess required skills, abilities or knowledge when joining the tech industry**: a lot of entry-level jobs actually require three years of experience [3]; Gaps between Engineering Education, and Practice (what an Engineer does in real life) do exist [14]; The software industry presents dissatisfaction in relation to the level of recently graduated professionals [12]; there is considerable room for improvement in what is taught to software students [in relation with job relevance] [9]; Many employers find that graduates and sandwich students come to them poorly prepared for the every day problems encountered at the workplace [4].

Some universities and programs even took steps to try and fix this problem in some specific classes by doing all kind of things: from purposely hindering and disrupting the software development processes [4], to adapting and incorporating industry training strategies into a software engineering course [12], to creating and adapting a project-based software engineering course that led the students to face with current, real-world engineering problems [6], and to highlight to students how relevant is having and developing critical soft skills to succeed in projects. [2].

At first, we thought that differences could come from different programs, so we explored the difference between Computer Science and Software Engineering programs, but those didn't prove really relevant; the official ACM/IEEE curricula [7] [1] are somewhat overlapping, and some studies trying to highlight differences in outcomes between CS and SE graduates were mostly inconclusive: a lot of core competencies are quite similar [11] [13]. And, those recently-updated curricula don't seem to incorporate lessons from the aforementioned efforts.

CHI'16, May 07–12, 2016, San Jose, CA, USA

ISBN 123-4567-24-567/08/06…$15.00

DOI: http://dx.doi.org/10.475/123_4

The acknowledgment of this skill gap and the efforts to train new graduates for the industry go back as far as 1992 [5]. So, **if in a quarter of a century little to nothing changed, what is the real matter**?

The hypothesis

So, we started thinking that maybe the problem is not one of implementation; it's not the university is unable to train graduates for the industry. We began thinking that, maybe, there is a **misalignment in incentives and expectations** for the various stakeholders: industry practitioners, university teachers, and students.

So, the phenomenal question is: what are the expectations and motivations for students, teachers in the field of computer science and software engineering? Following that, what do employers expect from new graduates in this field? Is there any gap between the expectation of these groups?

We have found some research on the topic: students usually do not have a personal vision for what they hope to do with a Computer Science degree, and there's a mismatch between what they are taught and what they had expected [8]; some initial work on understanding what makes a good software engineer was performed [10], but just on a few small samples from a very specific audience.

So, we would like to research into **what the expectations and motivations are** for our groups; we think we'll find a substantial **unalignment** between different groups' aims, that would justify the root reason for the perceived skill gap.

WHAT WE PLAN TO DO

We plan to **create a survey** where we ask questions to assess the thoughts of students, university teachers, and industry practitioners about:

- What should be the expected skills of fresh graduates (job proficiency, research proficiency, software writing, computational thinking, etc)

- What are the actual skills they encounter "in the wild"
- Whether they think gaps exist and should have been filled by somebody else

The survey will actually be tweaked depending on the target; some questions will be required for some groups, but not for others.

What we perceive is novel is that **we target all groups at the same time**, so we can put their answers in perspective, and that we will **aim for a larger group than previous studies**; we'd really like to reach 300+ answers to our survey.

In order to achieve such target, we'll create a website where we describe what we are doing, and we'll provide links to the survey and an easy way to share the website itself; we hope that a bit of advertising/social media push will help us getting to the numbers we need.

***** PROPOSAL TODO ITEMS *****

A feedback: What is the underlying educational problem or approach you see as a problem? What are the main problem(s) with this approach or problem? How can I fix it?

1- What are the research background that shows the motivation and expectations for student/teachers/employers in cs/se program?

2- what are the problems to limit the gap for new graduates in finding a new job after school? Why there is a gap in the new graduate skills and employer expectations?

3- what are some methodology that can shorten this gap? We can compare ivy league schools and other ones to see the differences and rate of employment after graduation?

OLD INTRODUCTION

One of the essential elements of a good software is to have a good software engineer (Paul Luo Li et al., 2015). The question is what makes a great software engineer? (Paul Luo Li et al., 2015) All different groups are looking into this question: employers want to hire a good software engineer, universities want to train a good engineer and new graduates want to become great (Paul Luo Li et al., 2015). Paul Luo Li et al. mention some of the employer's expectations for hiring software developers (Paul Luo Li et al., 2015). The research indicates that the expert engineers are more productive in terms of producing faster solutions, produce more amount of code in the same amount of time, and write code with much fewer bugs (Paul Luo Li et al., 2015).
Hewner and Guzdial investigate a game company on what are the employer expectations from new graduates (Hewner and Guzdial, 2010). They identify two of the essentials skills or expectations are high programming skills as well as people skills such as working in a team and collaborating with other people (Hewner and Guzdial, 2010). McConnell argue that software developers' personality traits like intellectual honesty, curiosity and being humble about their intelligence are important skills in addition to technical skills (McConnell, 2004). Hewner describes the mismatch between a student's expectations on skills they hope to learn and what they are taught in an introductory computer science class (Hewner, 2011). He notes that students come to the course with preconception about what they will learn in that computer science course (Hewner, 2011). The educators mention some of the students preconceptions as below (Hewner, 2011):
- Students expect to learn "advanced features" in application softwares.
- They expect to do IT work such as assembling computers from parts and configure routers.
- They expect to learn only about programming and not the architecture and theory.

Teaching computer science is different from teaching other subjects (Guzdial, 2014, https://cacm.acm.org/blogs/blog-cacm/174930-what-it-takes-to-be-a-successful-high-school-computer-science-teacher/fulltext) Good teachers should be able to read the code and help students to write code by hand off from computers (as well as at the computer) (Guzdel, 2014). On the other hand, the less successful teachers focus heavily on assessments and readings (Guzdel, 2014).
The technology and computer science industries are growing so fast (Ayofe and Ajetola, 2009). Therefore, the companies are looking for the graduates, who are able to use the latest technologies. However, the companies criticize the universities curriculum doesn't meet the practical issues in industry (Ayofe and Ajetola, 2009).

REFERENCES

1. Mark Ardis, David Budgen, Gregory W. Hislop, Jeff Offutt, Mark Sebern, and Willem Visser. 2015. SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *Computer* 48, 11 (nov 2015), 106–109. DOI: http://dx.doi.org/10.1109/mc.2015.345

2. Maria Cecilia Bastarrica, Daniel Perovich, and Maira Marques Samary. 2017. What Can Students Get from a Software Engineering Capstone Course?. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. IEEE. DOI: http://dx.doi.org/10.1109/icse-seet.2017.15

3. Kushal Chakrabarti. 61 percent of Entry-Level Jobs Require 3 Years of Experience. (????). https://talent.works/blog/2018/03/28/the-science-of-the-job-search-part-iii-61-of-entry-level-jobs-re

4. Ray Dawson. 2000. Twenty dirty tricks to train software engineers. In *Proceedings of the 22nd international conference on Software engineering - ICSE '00*. ACM Press. DOI:http://dx.doi.org/10.1145/337180.337204

5. R.J. Dawson, R.W. Newsham, and R.S. Kerridge. 1992. Introducing new software engineering graduates to the 'real world' at the GPT company. *Software Engineering Journal* 7, 3 (1992), 171. DOI: http://dx.doi.org/10.1049/sej.1992.0018

6. David Delgado, Alejandro Velasco, Jairo Aponte, and Andrian Marcus. 2017. Evolving a Project-Based Software Engineering Course: A Case Study. In *2017*

IEEE 30th Conference on Software Engineering Education and Training (CSEE&T). IEEE. DOI: http://dx.doi.org/10.1109/cseet.2017.22

7. ACM Computing Curricula Task Force (Ed.). 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc. DOI: http://dx.doi.org/10.1145/2534860
8. Michael Hewner and Mark Guzdial. 2011. How CS majors select a specialization. In *Proceedings of the seventh international workshop on Computing education research - ICER '11*. ACM Press. DOI: http://dx.doi.org/10.1145/2016911.2016916
9. Lethbridge. A survey of the relevance of computer science and software engineering education. In *Proceedings 11th Conference on Software Engineering Education*. IEEE Comput. Soc. DOI: http://dx.doi.org/10.1109/csee.1998.658300
10. Paul Luo Li, Andrew J. Ko, and Jiamin Zhu. 2015. What Makes a Great Software Engineer?. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE. DOI: http://dx.doi.org/10.1109/icse.2015.335
11. F. Meziane and S. Vadera. 2004. A comparison of computer science and software engineering programmes in English universities. In *17th Conference on Software Engineering Education and Training, 2004. Proceedings.* IEEE. DOI: http://dx.doi.org/10.1109/csee.2004.1276512
12. Carlos Portela, Alexandre Vasconcelos, Sandro Oliveira, and Mauricio Souza. 2017. The Use of Industry Training Strategies in a Software Engineering Course: An Experience Report. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. DOI:http://dx.doi.org/10.1109/cseet.2017.16
13. Ghulam Rasool and Touseef Tahir. 2014. A Comparison of Software Engineering and Computer Science Undergraduate Programs in Pakistan. In *2014 12th International Conference on Frontiers of Information Technology*. IEEE. DOI: http://dx.doi.org/10.1109/fit.2014.27
14. Vijayakumar Sivanesan. Analyzing the Gaps between Engineering Education and Practice. (????). https://gatech.instructure.com/files/638493/download?download_frd=1 EdTech Spring 2017.