

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

Alan Franzoni

Georgia Institute of Technology
Trieste, Italy
alan.franzoni@gatech.edu

Hasti Ghabel

Georgia Institute of Technology
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 [?]; Gaps between Engineering Education, and Practice (what an Engineer does in real life) do exist [?]; The software industry presents dissatisfaction in relation to the level of recently graduated professionals [?]; there is considerable room for improvement in what is taught to software students [in relation with job relevance] [?]; Many employers find that graduates and sandwich students come to them poorly prepared for the every day problems encountered at the workplace [?].

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 [?], to adapting and incorporating industry training strategies into a software engineering course [?], to creating and adapting a project-based software engineering course that led the students to face with current, real-world engineering problems [?], and to highlight to students how relevant is having and developing critical soft skills to succeed in projects. [?].

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 [?] [?] 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 [?] [?]. And, those

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 123-4567-24-567/08/06...\$15.00

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

recently-updated curricula don't seem to incorporate lessons from the aforementioned efforts.

The acknowledgment of this skill gap and the efforts to train new graduates for the industry go back as far as 1992 [?]. 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 [?]; some initial work on understanding what makes a good software engineer was performed [?], 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. So, we think that, most probably, employers will say they'll expect new graduates to be able to perform most real-world tasks; possibly, some students will say the same; but we suspect that most teachers won't answer this way, and they'll just say that they want their students to learn "computational thinking" or be prepared for doing research.

We think, as well, that employers still find some usefulness in new CS/SE graduates, or they wouldn't hire them, and possibly students would not enter such kind of classes.

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.

Then, we'll collect the data, graph and analyze it, as well as creating a final presentation to sum up what we'll find.

TASKS

Below, we listed the tasks that needs to be completed for this project. We displayed the number of hours for each task in front of it inside the parenthesis.

Milestone 1

Task 0: Collaborate with teammates and complete the **project proposal**. (15 h)

Task 1: Create a **website** that explains our project objectives to the readers and links to our different surveys. This website is required to enlarge the audience with different experiences in this field of computer science. (15 h)

Task 2: Plan the **survey questions** that target different categories including students, teachers and professors, employers, and recruiters. The survey questions are very important to lead the research to its goal. Correspondingly, the given answers from different categories can provide a good and promising results analyses. (12 h)

Task 3: Using **proper tools** to provide the survey. Some available tools that we consider are Google Docs, or Monkey surveys. (8 h)

Milestone 2

Task 4: Make the **survey** available to a smaller sample of categories. This gives us the opportunity to better understand

what we missed in our survey or what was extra, which does not give us important information. (10 h)

Task 5: The next step is to **validate** and **update** the survey questions based on the result and feedback from small group survey. (18 h)

Task 6: Create a plan on how to send our results to the people who are interested to know what we find in our research. We can setup a **mailchimp mailing list**. (10 h)

Task 7: Share the **validated survey with a large group of people**. We need to share it on different forums, blogs, linkedIn, facebook, or maybe targeting some companies. (15 h)

Final Steps

Task 8: **Collect the results** from our survey. Research on available tools that we can analyze our results and display them on a graph to effectively describe our findings. (20 h)

Task 9: Put all our findings together and **prepare the final project paper**.

Task 10: Provide the **final project presentation**. (20 h)