Proposal:
Discovering 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 [3]; gaps between Engineering Education, and Practice (what an engineer does in real life) do exist [12]; the software industry presents dissatisfaction in relation to the level of recently graduated professionals [10]; there is considerable room for improvement in what is taught to software students [in relation with job relevance] [8]; 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 [10], 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 different outcomes 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 for Computer Science [7] and Software Engineering [1], which many universities base their program on, 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 shared [9] [11]. And,

those 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 [5]. So, **if in a quarter of a century little to nothing changed, what is the real matter**?

RESEARCH QUESTION

We have two sets of questions. In the first set, our main question is: **does the perceived skill gap in fresh graduates exist because the academy is unable to provide a good training, or just because a) the academy is not even trying to do that kind of job, and b) the industry is taking that kind of job for granted, or c) the students think they should be getting something that the university has no intention to provide them with?** Here, we ask what could be the reasons that there exist a gap between students, professors, and industry professionals' expectations. We will look into these questions for both **undergraduate** and **post-graduate** level students that are recently graduated from school. We also ask the question that **how the students' degree can improve the chance of getting hired? Do the graduate-level studies help the students to gain adaptive skills in industry more quickly?**

In the second set of our questions, we are asking **what would be the best solutions that bring the university and industry's objectives closer to each other?**

THE HYPOTHESIS

We think that:

- Most students, when picking their major, have little to no idea what they're going to actually study, and they probably expect to learn mostly about programming and creating applications;

- Most teachers, when designing their courses, think about teaching what they deem useful to achieve the so-called *computational thinking* in their students;

- Most professionals and employers, when hiring fresh graduates, expect they'll be able to immediately and fully carry out whatever real-world task is assigned to them.

So, we provide two sets of hypothesis to approach the research questions.

Hypothesis 1: One of the reasons for the perceived skill gap is that all those that should - in an employer's view - care for learning some skills to be used at work, don't actually have that aim during their education phase. In other words, the misaligned expectations between industry and university causes the skill gap. Those skills have impact on job proficiency and possibilities to get hired. We think that the expectations among four groups differ. These four groups are composed of **undergraduate-level students, post-graduate-level students, educators and school staff, and industry professionals**. The main problem is not only that one side hasn't enough resources or skills to achieve a certain goal, but, rather, that there's a different vision or gaol on what should be done, and different and unaligned *rewards* exist for different groups.

Hypothesis 2: We think that the following hypothesis is that the graduate-level studies, on the contrary, can play an important role on reducing the skill gap between industry and university, and therefore, can be considered as one good resolution for that issue. More specifically, we think that **high-quality online graduate-level programs** such as Georgia Tech OMSCS (Online Master of Science in Computer Science) are quite aligned with industry expectations. These such programs also target many people from all over the world, who can become proficient in their job quickly as well as being active in an academic environment.

RESEARCH METHOD

It would be great to have one-on-one interviews with a lot of people, and ask them what were their expectations before starting their degree and after getting it, to understand professionals' point of view, and to try to understand what teachers and university staff designing courses is actually trying to do; but, that would involve an important amount of time, both for the researchers and all the people involved.

Still, we think we can leverage modern technology to achieve something useful; we plan to **create a survey** where we ask questions to assess the thoughts of students (post-graduate and undergraduate level), university teachers, and industry professionals to discover what it is, in their opinion, the current goal of university degrees (both Bachelor's degrees and post-graduate degree in Computer Science, Software Engineering or whatever a similar degree is called in one's country), and how that affects job proficiency and chances of being hired. Following this, we would like to find possible solutions on how to reduce the skill gap caused by misaligned expectations. In this step, we plan to provide more open-ended questions, where participants can describe their thoughts in more detail. In our questions, we will also point out graduate-level online programs and ask them what would be the benefits of such programs in resolving the described issue.

In summary, we hope to explain some of of the misaligned expectations among universities and industry. Following that, we would like to describe some possible solutions to resolve this phenomena.

Independent variables:

- Category: CS/SE undergraduate student, CS/SE post-graduate students, CS/SE professor/teacher/university staff, industry professional in the tech/software field;

- Age;

- Country of residence;

- Country where somebody got his/her degree (if any);

- Highest completed educational degree;

- Previous programming skills or actual job experience in the tech/software field before starting university;

- If employed, company size and tech department size;

Dependent variables:

All the variables are about a CS/SE bachelor's/post-graduate's degree fresh graduates.

- Self-reported current goal for bachelor's/post-graduate's degree: what they think that university aims at, right now;

- Self-reported ideal goal for bachelor's/post-graduate's degree: what they wish that university would do, if different from current;

- Self-reported expected GPA relevance for actual job proficiency;

- Self-reported university ranking relevance - how the ranking of a well-known university can impact the job qualification?

- Self-reported expected (or actual, for professionals) time to achieve full job proficiency when entering the industry?

- Self-reported expected time to land the first job for a current fresh graduate?

- Self-reported chances of getting hired for new graduates with no previous working experience (both post-graduate and undergraduate levels);

- Self-reported top skill which they think useful at a job that can't be taught at school (if any);

- Self-reported expected proficiency of graduates versus practitioners with a relevant work experience in the same ballpark (4 years) but without a degree, when getting a job;

- Self-reported expected proficiency of graduates versus practitioners with a relevant work experience in the same ballpark (4 years) but without a degree, after one years of being hired;

- Self-reported possible solutions to reduce the expected skill gap between industry and universities for new graduates;

- Self-reported role of online high-quality graduate level programs in reducing the skill gap - how programs like Georgia Tech OMSCS (Online Master of Science in Computer Science) can help to solve this issue?

Based on item 9th and 10th of *dependent variables*, we try to discover whether the perceived gap could actually be "reversed" - so, whether the graduates think they're up for a slow start when compared with practitioners, but are then able to gain the same (or more) speed.

What we perceive is novel is that **we target all categories 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.

RECRUITMENT
Since we would like to get a good sample, 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 will put our results and the provided graphs from our data analysis on the website to present our work. If anyone is interested to know the results, they can subscribe to our website and they will get the result explanations at the end of the research period. We will use SurveyMonkey or Google Forms as the tool to put our survey questions in it and collect the results. We might also use social media to get to the numbers of participants we need.

We think this research is relevant, since many of our peers giving us feedback thought it was very interesting, so we'll ask our classmates to share the website as much as they can, and we'll ask each survey-taker to share it as well, especially to categories that are different than himself (e.g. "Share this survey with your professors!")

The main purpose of our website is to easily share the goal and description of our research with many people, provide the survey link in the website, and make it a platform that we can share and present our results with whoever is interested to see the results.

DATA ANALYSIS
Once we collect the data, we will split our data into two parts. The first part will focus on investigating the expectation gap and the second part is focusing on finding the solution to reduce the skill gap. We will use **quantitative** and **qualitative** analysis to describe the results. We will take the quantitative approach for the first part of our investigations. We will use analytical tools (e.g. Excel or coding) to analyze our data. We'll try to uncover patterns that highlight the relations between some of our variables. We will use R programming to analyze our data. Here, we shortly describe the methods we will use to analyze our survey data:

1. Use R programming

2. Calculate the simple statistics for our variables (Mean, Max, etc.)

3. Create graphs for each of the questions with error bars

4. Create appropriate tables and diagrams to use

5. Provide analytical methods such as t-test and ANOVA to verify relationships and trends in our data

In the second part of our survey data analysis, which is focusing on finding best solutions to the issue, we qualitatively analyze the open-ended questions. This qualitative analysis helps us to understand the motivations and opinions on solving the problem. We will start reviewing the data as soon as the first pieces of data are collected. Collecting the data for this section might need more time, since it is based on participants' interpretation and explanation to each of the questions. This need a thorough review and understanding of their explanations to make sure we are not going toward false positives or false negatives.

LIMITATIONS
Threats to validity, both internal and external: we hope to get a good sample from around the world, but there's always the risk we incur in some biases - e.g. if some universities or organizations get really involved with the research we may get too many results from those specific institutions. We may also get a skewed number of answers from a specific target group - we'll need to understand what a good proportion is between groups as well, because we doubt the world offers the same amount of professionals, teachers, and students.

Additionally, there will be a limitation on finding a large number of participants that meet our group categories. It is more challenging to find a large number of teachers, professors, and also recruiters, since they have a busy schedule or they may not be allowed to participate in certain surveys due to their company policies.

Another thing is time limitation. We only have 4-5 weeks to provide the survey questions, send it to the people (small and large group), validate the questions, and analyze and summerize the results. Sometimes, it takes **a couple of weeks** for each participant to only complete the survey and collect the results.

Also, we may be getting skewed answers with respect to institutions; for example, we may get a lot of answers from students from a certain university, some teachers from a different one, and then we may get a lot of practitioners that had no contact with such universities, and hence have no such perception; it would be great to match students and teacher from certain institutions, and then let see what employers and professionals say about such students; but this would require quite more time and a far more complex setup than what we can afford for this research.

DELIVERABLES
We collect the deliverables and their due dates in the table below.

Deliverable	Due Dates (year 2018)
Five weekly status reports	Every week, 06/25 - 07/23
Intermediate milestone 1	07/2
Intermediate milestone 2	07/16
Final presentation	07/30
Final project	07/30

Table 1. Schedule of each deliverable

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.

a: Project Proposal

Task a1: Collaborate with teammates and complete the **project proposal**. (15 h)

Task a2: Collect the results we have so far and provide the **milestone 1**. (2 h)

Task a3: Collect the results we have so far and provide the **milestone 2**. (4 h)

b: Survey

Task b1: Learn how to create a productive survey. (4 h)

Task b2: Plan the **quantitative survey questions** that target different categories including students, teachers and professors, industry professional and employers. 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. (10 h)

textit*Task b3*: Plan the **qualitative survey questions** that provides open-ended questions. (8 h)

Task b4: Evaluate available online survey tools. (2 h)

Task b5: Using **proper tools** to provide the survey. Some available tools that we consider are Google Docs, or Monkey surveys. (3 h)

Task b6: 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. We are planning to share the survey at this level with our classmates at Educational Technology - Summer 2018. We will use Piazza platform to share our survey with this group of people. (1 h)

Task b7: **Validate** and **update** the survey questions based on the result and feedback from small group survey. (4 h)

c: Website

Task c1: **Set-up the website**. This website is required to enlarge the audience with different experiences in the field of computer science and software engineering. (8 h)

Task c2: **Provide the content in the website** that explains our project objectives to the readers and links to our different surveys. (4 h)

Task c3: Describe the results on the website. (4 h)

Task c4: Provide the graphs on the website. (4 h)

d: Data Analysis

Task d1: **Refresh our knowledge** regarding analysis methods. (12 h)

Task d2: **Collect** and **analyze** the quantitative data. (15 h)

Task d3: **Collect** and **review** and **analyze** the qualitative data. (20 h)

Task d4: **Apply analytical tools** and run analytical methods such as ANOVA and t-test to analyze the data. (10 h)

Task d5: Take **manual look** at the data and see if we can find any interesting pattern or results. (10 h)

r: R programming

Task r1: **Learn R programming** to use in our analysis. (10 h)

Task r2: Apply **R programming** to analyze our data. (20 h)

e: Share Data

Task e1: Share the **validated survey with a large group of people**. We need to share it on different forums, blogs, linkedIn, facebook, OMSCS slack channel, or maybe targeting employers and professionals in some companies. (5 h)

f: Final Steps

Task f1: Collect all the data together and make them ready for the final paper. (5 h)

Task f2: Put all our findings together and **prepare the final project paper**. (12 h)

Task f3: Complete the **final project** and **final project presentation**. (10 h)

Milestone 1

We plan to complete below tasks in Milestone 1:
Task a2, Task b1, Task b2, Task b3, Task b4, Task b5 for the first sample survey, Task b6, Task c1, Task c2, part of Task d1, part of Task d3 (We will apply some analysis on the results from our first sample survey sent to the smaller sample group).

Milestone 2

We plan to complete below tasks in Milestone 2:
Task a3, Task b5, Task b7, Task d2, Task d3, Task d4, Task d5, Task r1, part of Task r2, Task e1.

WEEKLY MILESTONES

Here, we describe the weekly due dates, goals, project-related assignments, and associated tasks due dates.

Week #5

- June 11, 2018 - June 18, 2018:

  - **Goal**: Complete the project proposal.
  - **Assignments Due**: Project Proposal
  - **Tasks**: Task a1

Week #6

- June 18, 2018 - June 25, 2018:

  - **Goal**: Set up the website, provide survey questions.
  - **Assignments Due**: Weekly Status Check 1
  - **Tasks**: Task b1 & Task b2 & Task b4 & Task b6 & Task c1

Week #7
- June 25, 2018 - July 2, 2018:

    - **Goal**: Collect and analyze the sample data, get feedback, work on the website.

    - **Assignments Due**: Weekly Status Check 2, Intermediate Milestone 1

    - **Tasks**: Task a2 & Task b3 & Task c2 & part of (Task d1 & Task d3)

Week #8
- July 2, 2018 - July 9, 2018:

    - **Goal**: Validate survey questions, Learn analytical methods and learn R programming.

    - **Assignments Due**: Weekly Status Check 3

    - **Tasks**: Task b7 & Part of Task d2 & part of Task d3 & Task d5 & Task r1 & Task e1

Week #9
- July 9, 2018 - July 16, 2018:

    - **Goal**: Collect and analyze the data, Apply R Programming.

    - **Assignments Due**: Weekly Status Check 4, Intermediate Milestone 2

    - **Tasks**: Task a3 & Task d4 & Task d5 & Task d2 & Task d3 & part of Task r2

Week #10
- July 16, 2018 - July 23, 2018:

    - **Goal**: Continue on analyzing the results with R programming, prepare them for the final paper and presenting them on the website.

    - **Assignments Due**: Weekly Status Check 5

    - **Tasks**: Task c3 & Task c4 & Task f1 & Task d2 & Task d3 & Task r2

Week #11
- July 23, 2018 - July 30, 2018:

    - **Goal**: Complete the Final Project and Final Presentation

    - **Assignments Due**: Final Project, Final Paper, Final Presentation

    - **Tasks**: Task f2 & Task f3

MEMBERS OF THE TEAM
The name and GT username of the team members are mentioned below:

1. Alan Franzoni : *afranzoni3*

2. Hasti Ghabel : *hghabelzadeh3*

TASK DIVISION
The tasks are divided among the team members. The number of each task is described in front of the name of each team member.

1. Alan Franzoni Task #s: a1, a2, a3, b1, b2, b3, b7, c1, c2, c3, c4, d1, d2, d3, d4, e1, f1, f2, f3.

2. Hasti Ghabel Task #s: a1, a2, a3, b1, b2, b3, b4, b5, b6, b7, c4, d1, d2, d3, d5, e1, f1, f2, f3

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. K. Chakrabarti. 2018. 61 percent of Entry-Level Jobs Require 3 Years of Experience. (2018). `https://goo.gl/u6wKSc`

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. 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`

9. 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`

10. 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`

11. 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`

12. Vijayakumar Sivanesan. Analyzing the Gaps between Engineering Education and Practice. (????). `https://gatech.instructure.com/files/638493/download?download_frd=1` EdTech Spring 2017.