Misaligned Expectations: Uncovering Different Aims in Universities and Industry

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 programs **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].

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**?

We started thinking that the matter was not an inability of the academy to properly train students and that, instead, there is a *misalignment* in the expectations of the university, industry, and students; each one goes by its own way and ignores others' desire.

With our research, we'd like to answer some questions:

- Could one of the reasons for the perceived skill gap be 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?
- Do such abilities have an impact on job proficiency and graduate hireability?
- Does the behaviour towards industry-related skill significantly change between undergraduate and graduate education, and can high quality graduate education programs (like OMSCS) help in bridging the skill gap?

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

RELATED WORK

Some universities and programs 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].

We thought, as well, 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.

METHODOLOGY

In order to understand what could be the reasons for the situation we've described, we **created a survey** where we asked questions to assess the thoughts of various categories related to our research - Computer Science and Software Engineering students (graduate and undergraduate level), CS/SE university teachers, and industry professionals in the tech sector - to discover what it is, in their opinion, the current goal of university degrees (both Bachelor's degrees and Master's 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 provided open-ended questions in the last section of the survey that participants would share their thoughts on possible solutions on how to reduce the skill gap caused by misaligned expectations. The survey questions were designed to highlight contrasts between what students achieve and what people would like them to achieve. In this last section, we also pointed out graduate-level online programs as a possible solution and asked them what would be the benefits of such programs on resolving the described issue.

Beyond the category, we had some other independent variables, like age, country of employment, country where people got their degree, highest completed education degree, company size.

We tried to reach out as many people as possible but, while we had an internal target of at least 300 respondents, we just got about half that number of responses. We leveraged a dedicated website, https://www.misalignedtech.com, in order to disseminate our research outside the OMSCS.

Once we collected their answers, which were constituted essentially by categorical data (for both dependent and independent variables) and some qualitative data (open questions), we looked at differential patterns: are there situations where some variable, especially the category, has a serious impact on some perceived or desirable skill?

RESULTS

Surprisingly, we've found that the expectations don't seem to be that misaligned on many topics. All of our categories, for example, have got very similar opinions on most BS graduate skills and would-like skills with one interesting twist.

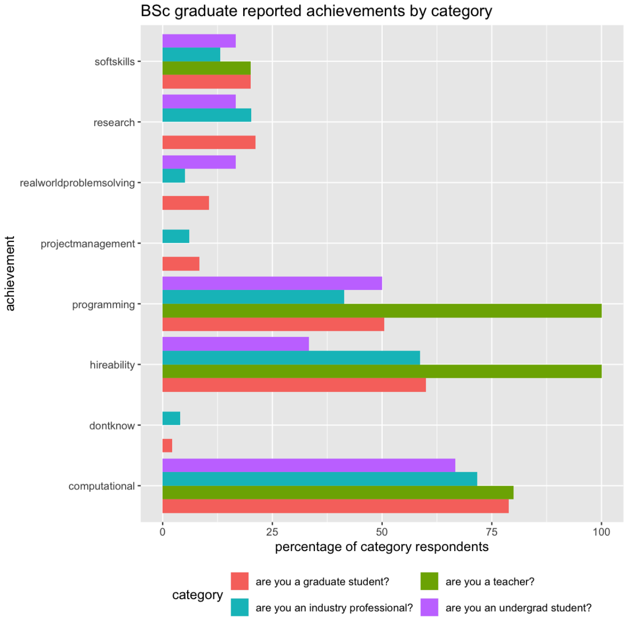


Figure 1. Currently, what do you think an UNDERGRADUATE student in CS or SE achieves, when he/she graduates?

We can see - figure 1 and 2 - that most of our respondents, regardless of their category, think that a BS graduate achieves the so-called *computational thinking*, and, secondarily, a certain chance of getting hired and some programming skills. Here, we can actually see a bit of contrast between two categories: the teachers, and the industry professionals. The former group think that programming is something a BS graduate definitely acquires; the latter, being the ones that actually employ such skills, are far more reluctant to say that you get this skill in university.

Regarding the would-like skills, there are not great surprises; it seems that the most interesting selected items, that aren't achieved at university, are real-world problem solving, soft skills, and project management. Undergrads especially did not vote for this latest skills, however, improvement on chances of getting hired is one of this category's would-like achievements from academic educations.

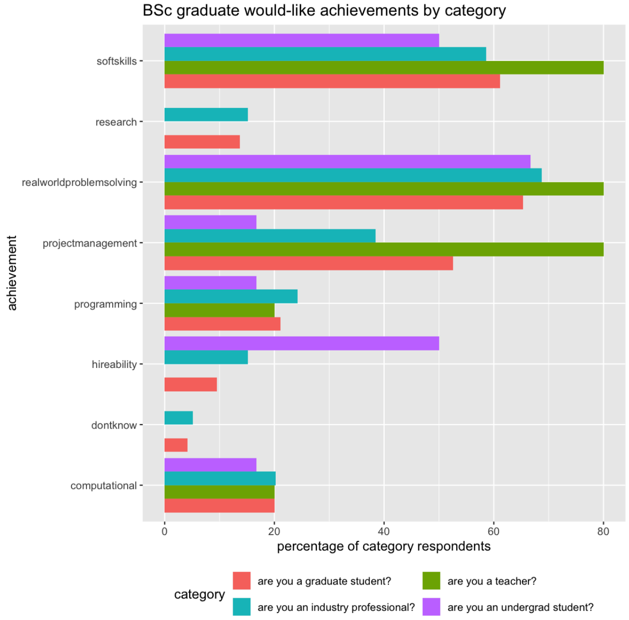


Figure 2. Are there any skills, that you think an UNDERGRADUATE student would need to learn at university, where it is not taught by universities, and are essential in working industry?

One surprise comes from programming skills for undergrads. It's not marked as a clear achievement by most respondent, and **neither it is marked as a desiderata**. How should somebody hone his programming skills? On the other hand, it seems that programming skill is mostly achieved by post-graduate level students and is not selected as a high demand for would-like achievements.

The other surprise comes from research skills. Teachers don't think it's something that is taught, and neither that should be taught. Possibly, that's left to graduate education?

Here we see the opinions about graduate education. It appears to be a partial solution for some parts of undergrad education, but not completely. Most people *but industry professionals and graduate-level students* would agree that MS education improves your programming skills; since industry professionals are actually the most qualified at such opinion, it's not really encouraging, since they estimate MS graduate abilities on par with BS graduate abilities. On the other hand, teachers and undergraduate students categories ($> 75\%$) selected this skill as an achievement after MS graduation.

We see better results for research: a reasonable amount of respondents think that graduate education is the stage at which research skills are taught.

But real-word problem solving is still an open problem, even after a MS degree; and soft skills, along project management,

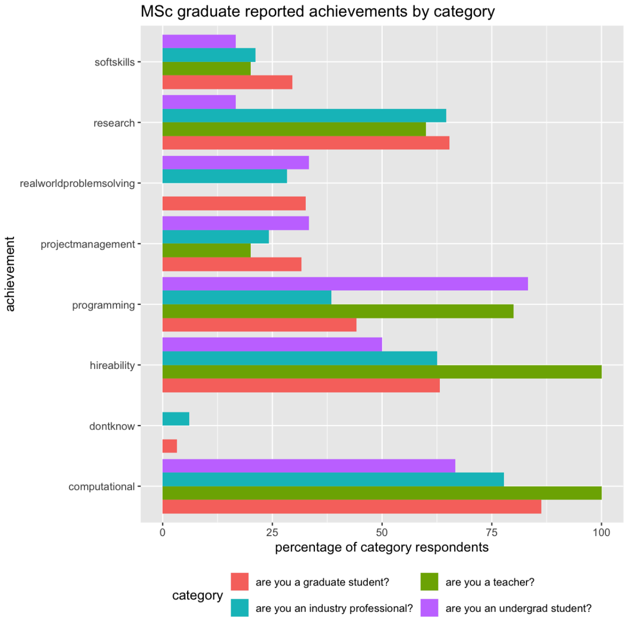


Figure 3. caption

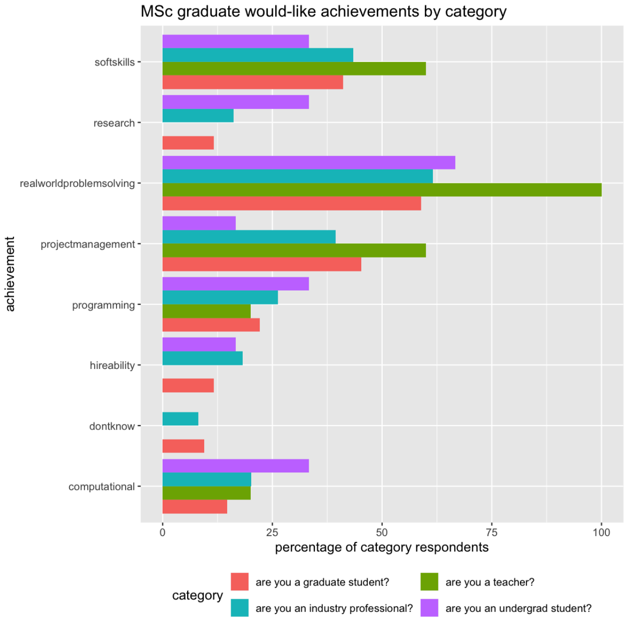


Figure 4. caption

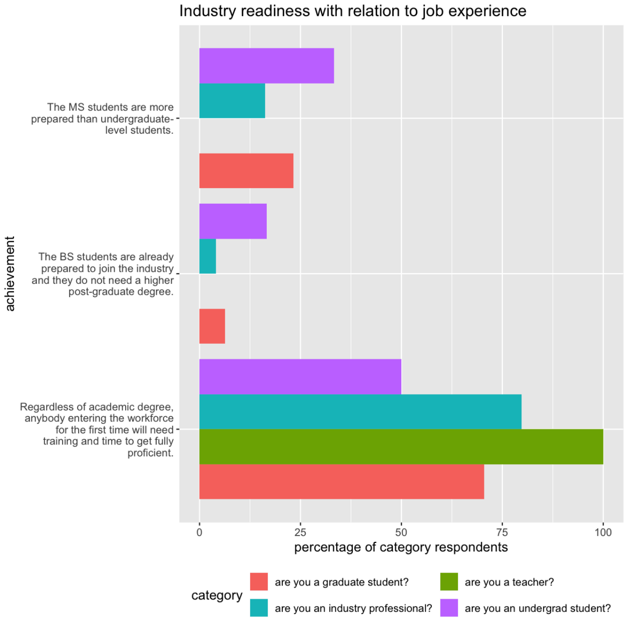


Figure 5. caption

could fare much better; there seem to be a large amount of teachers that agree on the fact that MS graduates would need more skills that can be applied directly in the real world.

Generally, we can see more traces of this belief in another answer:

Only a few undergrads (12.5%) hope that, at the end of their four-year studies, they'll have a high chance to enter the workforce and just have a great career. Other categories mostly believe that an apprenticeship phase will be mandatory for everybody - in this phase, most probably our graduates will learn about **soft skills, project management, real-world problem solving, and actual programming**.

This idea is reinforced by other answers in our survey; when discussing the chances of being hired, we asked our respondents to compare experienced and unexperienced candidates with and without degrees.

As we can see, a large majority (between 75% to 80%) thinks that experience matters a lot. Interestingly, teachers seem that experience is even more important, probably acknowledging the industry-school expectation gap and our supposed misalignment.

One last interesting result explains the job proficiency in short-term and long-term for both BS and MS graduates:

Here, we can explain two things:

- In the short term, experience is more effective than a degree.
- In the long term, a degree doesn't necessarily mean a better worker.

While the first statement may be a bit expected, we think that the second is clearly a novelty; even teachers don't acknowl-

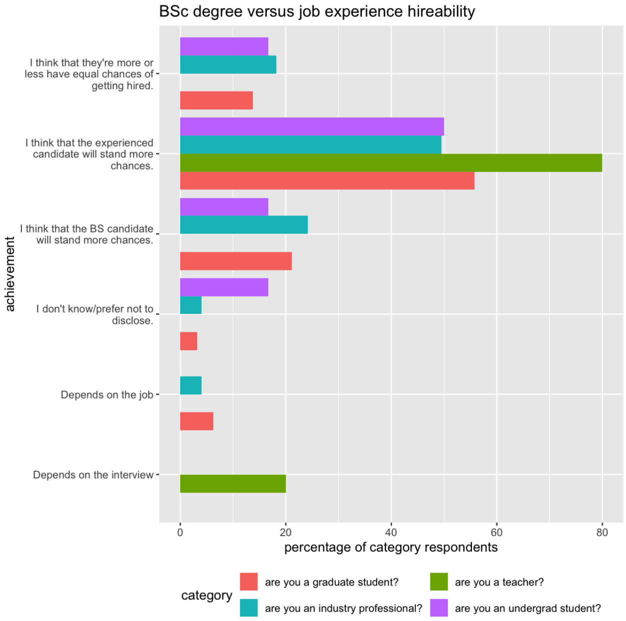


Figure 6. caption

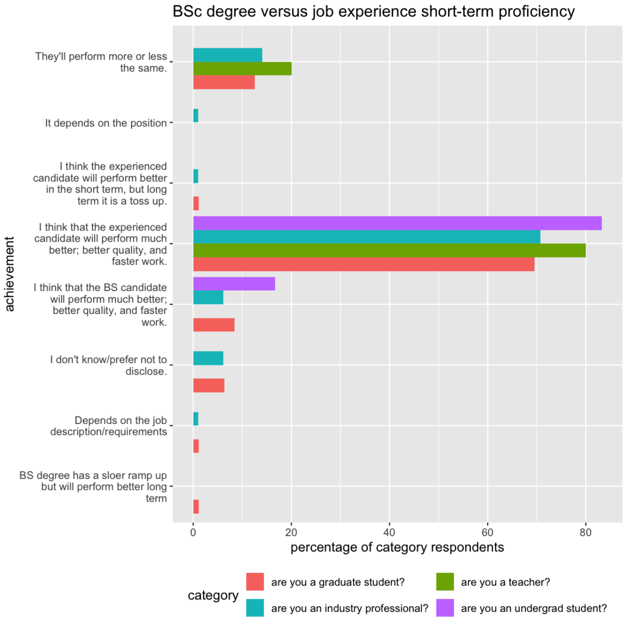


Figure 8. caption

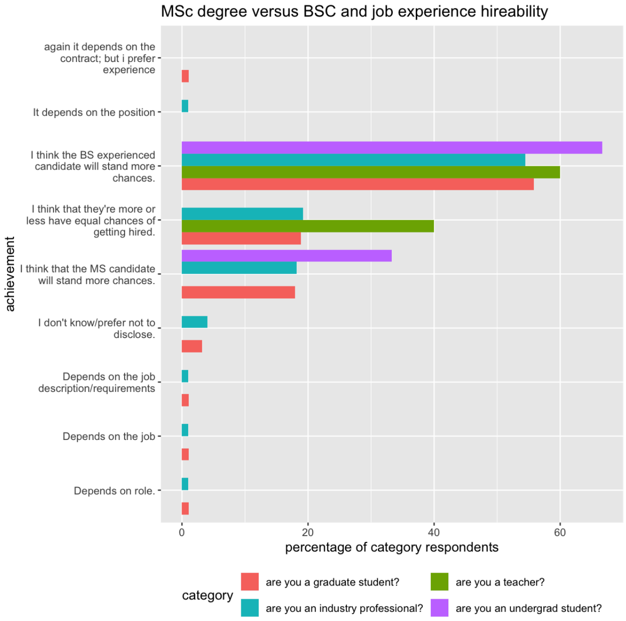


Figure 7. caption

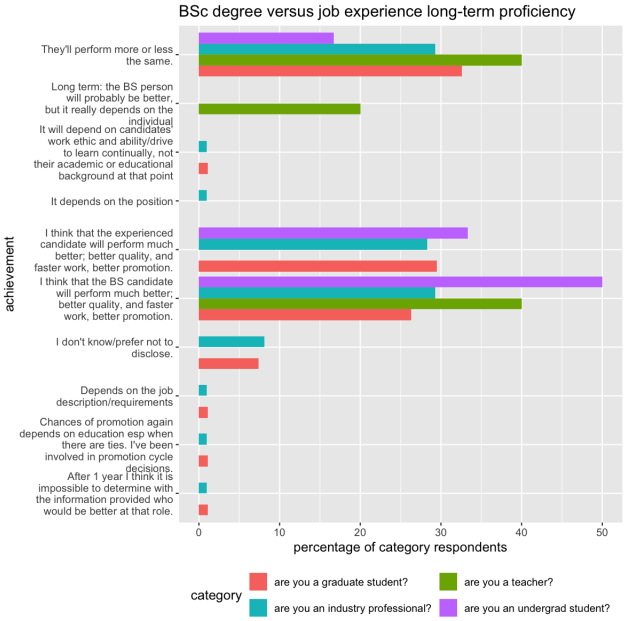


Figure 9. caption

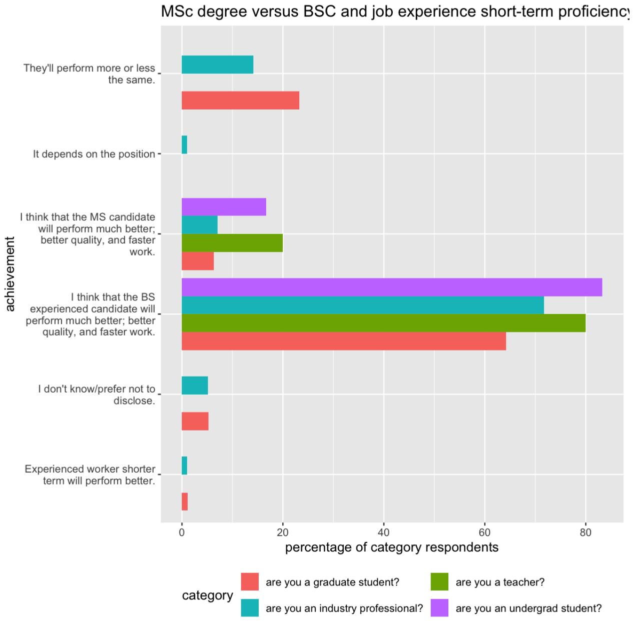


Figure 10. caption

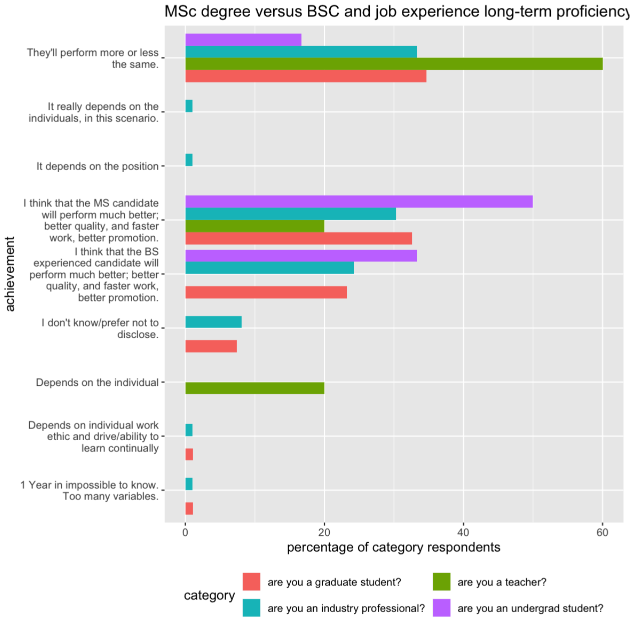


Figure 11. caption

edge a clear advantage for graduates at doing their job. This is quite a clear image of the misalignment: there seems to be at least a slight indication that people in the academy don't really think that a lot of the topics they teach are *that* aligned with modern industry requirements, neither in short-term nor long-term proficiency.

We have seen quite scattered responses about the impact of GPAs and top schools, and we don't think we can draw big conclusions about those topics (The graphs are collected at http://www.misalignedtech.com/)

In the next step, we collected data about possible solutions and suggestions on how the students in both levels can gain useful work experience during their studies. We found that most frequent answers are as internships, real-world work, part-time jobs, extra courses or learning new languages, open-source projects, research, and studying. As it shows in the table below, the majority of the participants pointed out "internships" as the most effective practice for that. The next two major suggestions are part-time job and open-source projects. It is interesting that the part-time job is selected doubled the time for MS studies compared to BS studies. Another item to point out is the Research selection. As it shows in the table, 13% of the respondents think that Research can help in MS studies to gain useful work experience. However, only 6% recommend Research as an option to better be prepared for job industry.

Here, we looked at valid alternatives to MS education for a professional in need of retraining. As we expected, the MOOC and, certainly, on-site job training received the highest percentage among all the categories. The "in-person learning" received less than 40% of responses from all categories except teachers and professors. about 60% of professors think that in-person studies can be a convincing option for MS studies. Another interesting item is to compare undergraduate-level students responses. It shows that there is a percentage increase (about 20%) of choosing MOOC compared to on-site job training. This shows the high demand from undergraduate students to attend high-level online graduate programs and substitute this option with in-person or on-campus studies.

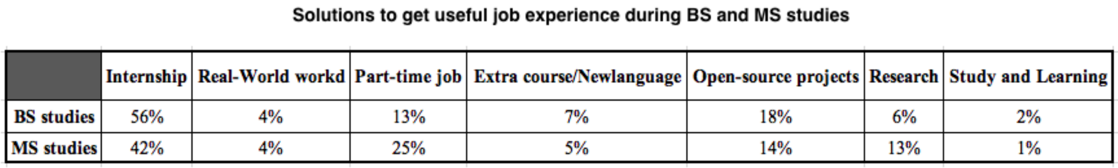
LIMITATIONS

We think our research has several limitations.

First: we didn't get as many respondents as we would have liked. We got about half the respondents we' were expecting; we miss statistical power.

Second: our categories are quite polarized. We have got a lot of industry professionals and graduate students, but we lack teachers and undergraduate students. Surely ,those few undegrads and teachers' opinion has a disproportionate effect in our analysis.

Third: most of our respondents come from the US. We don't think this research can have a worldwide validity, it's probably just one view of the problem.



Solutions to get useful job experience during BS and MS studies

	Internship	Real-World workd	Part-time job	Extra course/Newlanguage	Open-source projects	Research	Study and Learning
BS studies	56%	4%	13%	7%	18%	6%	2%
MS studies	42%	4%	25%	5%	14%	13%	1%

Figure 12. The table shows the suggested solutions by participants on how to gain useful work experience during BS and MS studies.

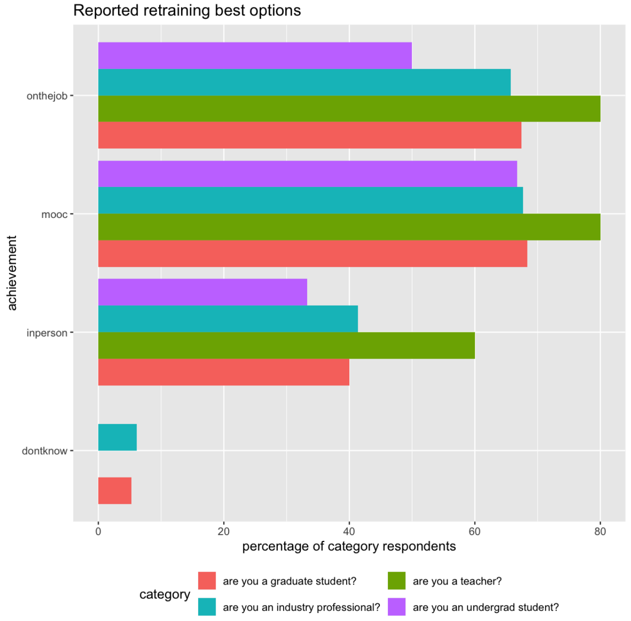


Figure 13. This graph displays the comparison between category responses on alternatives to MS education for a professional in need of retraining.

Fourth: we had initially missed (an error while converting a document to Google Forms) a fundamental question in our survey (the category). Since we knew some of the respondents, we have inferred the categories for our analysis' sake for a small set of initial respondents (the survey was later amended).

Many people (in online forums where we had spread our survey as well) told us one interesting point about the "chances to get hired" section: the experience, or the degree, only matter up to a point in order to be *hired*. It may matter in order *to get an interview*: if we ever replicate this survey, it would be better to substitute "hireability" with "chances to land an interview".

CONCLUSIONS

About our two main hypotheses, we could say that:

For the first hypothesis, we could say that collected data *partially supports* it. Undergrad programs aren't meant to teach project management, problem solving, or soft skills, while a lot of people would just love to see Bachelors graduate with such abilities; so, there's a misalignment between the intentions of the industry and of the researchers. Interestingly, most undergrad students seem fully aware of the situation.

Programming, on the contrary, seems a matter of teaching abilities. It seems that schools are unable to create good programmers. But then, most people just seem to think that apprenticeship and on-the-job training are not replaceable by pure education; maybe we should just accept that we're yet unable to abstract away that kind of learning from real-world experience, and we should scale down students (and employers!) expectations about new graduates: they won't be good programmers without an appropriate on-the-job training.

The graduate education part appears to be a bit more foggy. We supposed that graduate-level programs would better fill the school-industry gap, but we cannot say that this is the case. Most of the industry is not concerned with research, and most industry professionals don't see great programming skills in MS graduates. For sure, it's a beginning, it's something more than basic BS education; but, probably, spending the same amount of time on a real job would yield the same results about soft skills, real-world problem solving and project management.

POSSIBLE FUTURE WORKS

We think it would be very interesting to replicate the experiment on a different scale, with slightly different premises. **Recruitment** seems to be the hardest part of our research; it would be interesting to partner with some large organizations (be it companies, conferences, universities) in order to push a (similar) survey to them.

It would be interesting to investigate the concept of good programmers, as well. Do bootcamps/MOOCs/other kind of programs produce better coders than university? Are they on par, but within a shorter timeframe? Could the university improve something?

It may be interesting to check if multiple categories (e.g. industry professionals that are MS students as well) have significantly different answers from "bare" categories; this was a bit outside the scope of our analysis because of time constraints and because of the polarization of our respondents, but could be quite useful.

It would be very useful to get more data from the academy. We haven't seen a clear indication for the purpose of the higher education programs; we may say that they appear a bit confused on whether it would like to focus on research, or it would like to help more the industry, or would just pursue knowledge for knowledge's sake - which could be a totally valid purpose!

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.