What Skills Do CS Grads Need? Colleges And Employers Disagree.

Tyler Menezes
tylermenezes@codeday.org
CodeDay
Seattle, Washington

ABSTRACT

Many computer science students report feeling under-prepared for interviews and entry-level jobs, and colleges have been working to address this problem, but are we putting effort into teaching the right skills? This poster reports initial results from a survey of both CS professors and the industry representatives who hire and work day-to-day with our graduates. We find that, when asked what skills are most important and most lacking in today's graduates, industry and academia disagree.

ACM Reference Format:
Tyler Menezes. 2023. What Skills Do CS Grads Need? Colleges And Employers Disagree.. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3545947.3576369

1 INTRODUCTION

Most students studying computer science pursue their degree in order to improve their job prospects, but studies have shown that many recent CS graduates felt under-prepared for job interviews and the day-to-day work of a software developer. [1, 4]

Past efforts to better serve these students have involved the creation of advisory boards and industry working groups,[2, 5] but these usually comprise only high-level technical and HR leaders. [3]

We interviewed and surveyed more than 150 engineers involved in hiring and working with entry-level computer science graduates. We present a list of skills employers desire in recent graduates, with aggregate ratings of both importance for job function, and how prepared current graduates are. Finally, we compare these results to a survey of CS professors to show that a gap remains between what industry desires, and what colleges believe they must work to provide.

2 OVERVIEW

This study comprised two steps: synthesis of interviews to create a list of the skills which CS employers desire, and then surveying CS employers and college professors to create a prioritized list for each.

We recruited the representatives of CS employers from an pool of full-time software engineers who had recently volunteered to mentor CS students in a non-profit program. We limited our study group to only those who met both of the following requirements within the last year: contributed to a hiring decision for an entry-level CS position, and worked in a technical capacity with a new-graduate in an entry-level CS role.

We began by surveying a subset of these industry representatives and discovered the skills they desired for entry-level CS roles fell into four categories — Independent Software Engineering Process (the general ability to narrow down a problem, conduct research, experiment, and verify a solution); Interpersonal; Leadership; and Specific Technical Knowledge — which each had 4-22 specific skills.

We asked our full group of industry representatives and CS professors to rate each skill's "Importance" (how much it contributes to success in interviews and new-grad job performance), and "Ability" (how well new CS graduates currently meet industry expectations). We calculated each skill's "Impact" as its Importance times 1/Ability.

3 RESULTS

We recruited 153 industry representatives in total, of which 32 were interviewed to create the initial list. 6% represented small companies (less than 20 employees), 17% represented mid-sized companies (20-199 employees), and 77% represented large companies (200+ employees). We recruited 17 professors who represented bachelors CS programs from 16 colleges.

Overwhelmingly, CS professors believed the highest Impact skills were in the category of Specific Technical Knowledge (such as Git, APIs, or containers).

By contrast, employers did not prioritize Specific Technical Knowledge. The highest Impact skills according to employers — those they believed CS degrees most needed to address — were in the categories of Interpersonal and Independent Software Engineering Process, namely: (1) managing change and uncertainty, (2) identifying and defining problems using debugging techniques, (3) experimentation, learning by doing, (4) online and peer research to discover existing solutions to a problem, and (5) technical speaking/presentations.

REFERENCES

[1] Theresa Beaubouef, Wendy Zhang, Ghassan Alkadi, and Kuo-Pao Yang. [n.d.]. Beyond the computer science curriculum: empowering students for success. 26, 4 ([n. d.]), 7.
[2] Marisa Exter, Secil Caskurlu, and Todd Fernandez. [n.d.]. Comparing Computing Professionals' Perceptions of Importance of Skills and Knowledge on the Job and Coverage in Undergraduate Experiences. 18, 4 ([n. d.]), 21:1–21:29. https://doi.org/10.1145/3218430
[3] Tyler Menezes, Alex Parra, and Mingjie Jiang. [n.d.]. *Open-Source Internships with Industry Mentors.* https://arxiv.org/abs/2111.04414
[4] Damla Oguz and Kaya Oguz. [n.d.]. Perspectives on the Gap Between the Software Industry and the Software Engineering Education. 7 ([n. d.]), 117527–117543. https://doi.org/10.1109/ACCESS.2019.2936660 Conference Name: IEEE Access.
[5] Alex Radermacher and Gursimran Walia. [n.d.]. Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM technical symposium on Computer science education* (New York, NY, USA, 2013-03-06) *(SIGCSE '13).* Association for Computing Machinery, 525–530. https:

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576369

//doi.org/10.1145/2445196.2445351