Desired Qualifications Sought in Entry Level Software Engineers

Sid Stamm
stammsl@rose-hulman.edu
Rose-Hulman Institute of Technology
Terre Haute, Indiana, USA

ABSTRACT

Global demand for software engineers continues to strain the technology sector with unfilled software engineering positions and stiff competition for hiring developers who are on the market. To attract more candidates, technology firms have increasingly been working closely with universities to recruit new graduates to fill their jobs. Universities hoping to minimize mismatches in job placement for new software developers should teach the skills and attributes that enable entry-level software developers to succeed. This paper presents a case study of hiring demands at one large, well-established technology company that reveals the most sought-after attributes in new hires. We discuss results of our interviews with five software development hiring managers and the results from a wide survey of engineers and architects from various levels and experiences. The interviews and surveys reveal that some qualifications are widely desired, and others have varying demand based on functional area, technology, or type of development. Ultimately, professional skills (e.g., teamwork) and personality traits (e.g., strong initiative) top the list of desired attributes, along with a few fundamentally broad technical skills. One key takeaway is that candidates who learn professional skills from university programs may be more readily hired into their first software engineering job than those whose education focused mostly on technical areas.

CCS CONCEPTS

• **Social and professional topics** → *Software engineering education*; Employment issues.

KEYWORDS

Software Engineering, Professional Skills, Hireability, Employer-Sought Skills

ACM Reference Format:
Sid Stamm. 2023. Desired Qualifications Sought in Entry Level Software Engineers. In *Proceedings of the 54th ACM Technical Symposium on Computing Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569849

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.
SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569849

1 INTRODUCTION AND RELATED WORK

There is a wealth of research where investigators seek to identify precisely what makes software engineers successful; this is driven by both university programs (who wish to provide the best preparation for their students) and for software companies (who want to hire the right people for their jobs).

Much of the existing research focuses mainly on technical, or coding skills [3, 12], but there are indications from other studies [4] that skills that transcend technical domains like coding or software may be just as important. Sometimes considered "soft skills", the Accreditation Board for Engineering and Technology (ABET) and others refer to these as "Professional Skills" [2, 6, 8] and since 2000, ABET has emphasized their importance in their criteria. Software engineers, especially entry-level engineers, need these skills among others to have a successful career. Radermacher and Walia called this out in their 2013 literature review of this area [7] and Hewner and Guzdial [3] echo the industry's desire for these skills too.

In 2008, Begel and Nagappan showed that working with others is a critical skill [1], and since then many universities have included pair programming in their curriculum. This is, however, not the only professional skill needed to succeed as a software engineer.

In 2019, Valstar provided a detailed analysis of the gap between skills needed and skills desired, offering ways to structurally close this gap [10]. To confirm that Academia is on the right track, we hope to show the qualifications desired are similar to what a large technology company seeks. Expanding upon the work, Valstar et al. queried faculty to see what's stopping universities from closing this gap [11]. They find resource obstacles slowing the change needed in academic programs, which could benefit from data to support requests for funding or time. We hope our work contributes additional support for the urgency in adopting acdemic programs to the rapidly changing industry.

Lundberg et al. probed employers for "Skills, Knowledge, and Personal Attributes" that graduates from their programs need to be successful on the job after graduation [5] and found that in 2020 employers *still* wanted better communication skills than the examined programs offer their graduates.

The gap in communication skills is a clear example of how skills and attributes needed for a degree-holding job candidate to be "fully operational" (implicitly defined as the ability to work autonomously without additional training in [5]) includes more than the attributes and skills demonstrated by students who complete a particular degree program. Szynkiewicz et al. call this *employability* [9], and point out that employers often fill the gap between "employable" and "fully employable" (i.e., between "will hire" and "fully operational") with additional post-hire training. Extra training is commonly used to bridge this gap due to high demand for software engineers and although not ideal (costing money and time),

many companies find it easier to train them on the job than to find already *fully employable* candidates.

Lundberg et al.'s work illuminated one part of the employability gap: there are skills that can be learned after hire. There is an additional component of *employability* for job candidates that were turned down for the position.

> *We observe that the union of skills "trainable after hire" and skills "required for hire" is the set of qualifications degree programs should consider adding to their programs.*

Thus we sought to begin identifying what qualifications fall into this gap at one large employer to help guide improvements in many university programs.

Lundberg et al. [5] conducted follow-up interviews and a survey of many different employers, which is similar to this work and Hewner and Guzdial's [3], but Lundberg et al. focused on a few data points from many employers and targeted specific degree programs, whereas we elicited all qualifications from one employer for use in guiding improvements to or validation of many different degree programs. Our work is one attempt at establishing coordination between a large Fortune 100 employer and the university employing this paper's author to attempt uncovering *what qualifications* universities should be adding to their curriculum.

Li et al. [4] broadly surveyed engineers to identify top qualifications for *employability* at all levels; they also conducted interviews to interpret their results, concluding:

> *[T]he top five distinguishing characteristics of great engineers are writing good code, adjusting behaviors to account for future value and costs, practicing informed decision-making, avoiding making others' jobs harder, and learning continuously.*

These characteristics are a mix of technical and professional skills, but Li's work begins with a pre-set menu of qualifications and it's not clear whether these skills are used equally throughout an engineer's career. The result of Li's work is rigorous and deep, but does not address how university programs can better equip software engineers to get that first job where additional training may not be provided. Educators want to know what is missing from what they teach for an *entry-level* position, and thus what they can focus on that will ensure maximum *employability*.

1.1 Contribution

Ultimately, the question we aimed to answer through interviews and a survey was, *"What characteristics does an ideal entry-level software engineer possess?"*

Our work employed Hewner and Guzdial's methods [3] with a broader audience to help universities identify gaps in their curriculum. We aimed to find qualifications desired in software engineering job candidates so universities may find where *employability* requires something not covered in their programs' educational outcomes. In our study, we elicited qualifications from hiring managers (through individual interviews) and asked software engineers (via online survey) to rate qualifications in entry-level software engineers at one large Fortune 100 company (who wishes to remain unnamed in this work while rapidly expanding their software engineering capacity). Ultimately, we found that professional skills top the list of things that differentiate top-level engineers from the rest. We hope this data will help emphasize need and find opportunities for improvement in degree programs to make graduates of those programs more fully *employable*.

2 METHOD

We elected to conduct two rounds of one-on-one interviews with hiring managers to allow the participants to express opinions contrary to their peers or superiors without social pressure to conform to company-wide norms. We asked interview subjects to enumerate qualifications, then asked them to rank the qualifications. Lastly, we asked a wider set of engineers to rate the importance of top-ranked qualifications. This procedure is very similar to a technique used by Hewner and Guzdial [3] employing interviews to brainstorm qualifications and a survey to rank them. A key strength of using individual interviews over a focus group was the ability to identify unique rankings in various software focus areas (e.g., cloud computing or device firmware) from managers in those areas. Later, we were able to compare the separate rankings to survey results from engineers in those same areas.

This research involved human subjects, and was conducted under supervision of Rose-Hulman's Institutional Review Board via protocol number RHS0373. Individual subjects' identities were not retained, consent was obtained from each participant, and participation was entirely voluntary. Any notes or recordings created during this experiment were made with consent of the subjects and stored in digital form; they did not include subjects' names or identifying markers and and the notes/recordings from individual interviews were destroyed at the conclusion of the study.

In the first round of interviews, we met with five hiring managers and asked them to construct lists of qualifications they desired in entry-level software engineering candidates. This began with the interviewer asking each subject to identify the team's core area (cloud, firmware, DevOps, mobile, etc), and then estimate importance of hiring; to estimate "importance", subjects were asked to estimate the number of engineers they've hired in the past three years and how many they expect to hire in the future 12 months. Next, the interviewer asked each subject to identify qualifications or "attributes" desirable in entry-level software engineers. To help generate and guide the exercise, the interviewer asked the following questions:

- Characterize a "star performer" you have worked with, who was an entry-level software engineer.
- What were skills, behaviors, or abilities *present* in this "star performer" that contributed to their success.
- What were skills, behaviors, or abilities *absent* in this "star performer" that could have contributed to their success.
- Characterize a hypothetical "perfect new hire", who just completed a college/university degree program.
- In what areas do you want new hires to be *strongest* before you hire them?
- What skills, behaviors or abilities do you expect most entry-level engineers will need to learn *after they are hired.*

Once the first round of interviews was complete, we combined the lists of qualifications together into one list of qualifications, removing duplicates.

We began the second round of interviews by presenting the first subject with the combined list and a transcript of notes from their first interview. After ensuring all of their desired qualifications were included in the big list, the interviewer asked them to arrange the items in the list by importance. The interviewer acted mainly as a facilitator to help the subject employ a sorting method if necessary, but mainly encouraged the subject to talk through their reasoning and reorder using the method of their choice. After each subject completed their ranking, the updated list was used as the starting list for the next interview subject, but the subjects were not told how the initial list order was produced. This allowed other subjects to reflect on rankings from the other subjects without feeling pressured into agreeing with the other subjects.

During the ranking exercises, interview subjects were also encouraged to clarify any of the qualifications. For each qualification that was rephrased by an interview subject, all previous subjects were given the new phrasing and asked through email conversations to reconsider (verify or change) the position of the updated item in their ranked list. At the end of the second round of interviews, we had five ranked lists of qualifications. Each list was unique to each subject, and likely to their functional area.

We reordered the list of qualifications by the average of ranks chosen by the interview subjects. We discarded the lowest-ranked 11 qualifications (leaving 42) to avoid an excessively long survey and because the interview subjects indicated these were mainly "nice to have" and not very important.

The final part of our study involved gathering information from a broader set of participants. We asked a large group of software engineers to fill out an anonymous online survey that first identified the type of engineering they most frequently do (e.g., cloud, firmware, DevOps, etc), and then asked them to rate the qualifications. To make the survey more approachable, we split qualifications into three categories: Attributes, Professional Skills, and Technical Areas. Survey participants were presented with one category of qualifications at a time and asked to rate each qualification's importance on a scale of 1 (Not Important) to 5 (Very Important).

3 RESULTS

We requested interviews from a set of five subjects from various business groups across the company, and all of them agreed to volunteer for brainstorming and ranking of qualifications. Their levels of seniority varied from senior director to first-level manager, and their teams' focus areas ranged from DevOps, to Cloud, to broad "Labs" (all areas). Given this variance, we were able to look at segments of the survey responses in comparison to very focused hiring managers, and also compare the broad view of the "labs" managers to the survey results as a whole.

3.1 Interview Results

The five interview subjects came up with a total of 53 qualifications after duplicates were removed. During the interviews, the subjects were not coached on what types of qualifications were "fair game" for this list; we intentionally asked for "all qualifications" in the interviews so the subjects would pick out things we teach students in our universities, but also professional skills or things we may not teach them. As a result, many of the desired qualifications are not considered "Computer Science" by many scholars, and might instead be classified as professional skills or personality traits not limited to use in software engineering.

The five subjects were asked to rank the 53 qualifications and the majority chose "CS Fundamentals" as the most important. When pressed on the issue, many indicated they assumed a degree in computer science or software engineering assured this qualification, but it was an important qualification nonetheless. The other three most important qualifications identified by the five hiring managers included *Strong Initiative / Self-Directed*, *Problem-Solving Technique*, and *Teamwork Skills*.

13 (65%) of the qualifications ranked by managers in the top 20 are those that would fall into the six professional skills called out by ABET's EC2000 criteria [2] or are personality attributes like "curiosity". These are not necessarily a technical skill or domain knowledge within computer science. These are qualifications like *delivers what's promised* (professionalism), or *Knows when to stop and ask for help* (lifelong learning and teamwork).

Top-ranked computing qualifications included *Conceptual-Level Understanding of CI/CD Platforms*, *Debugging Skills*, and *Ability to apply Abstraction in Design/Implementation*.

3.1.1 Discussion. The average ranking of qualifications indicates consistency in what the subjects considered most important and least important; top-ranked qualifications were consistently ranked near the top by all managers, and bottom-ranked ones were consistently ranked lower. Items ranked in the middle had much more variability, suggesting that managers with different levels of responsibility or who oversee different types of focus (e.g., cloud computing) may value them differently. The consistency at the top and bottom indicates general agreement across functional areas for the extremes, and most importantly, the qualifications in highest demand by hiring managers.

3.2 Survey Results

The results of the survey ratings are shown in Table 3 ordered by average rating (highest first). 24 subjects from a variety of functional areas completed the online survey including Cloud (11), Desktop (5), Firmware/Embedded (5), Data Science (1), Other (1), and "All of the Above" (1). None of them self-identified as managers.

3.2.1 Discussion. The results of the survey indicate strong agreement about the most important qualifications, but there was a wide range of opinions on those ranked lower on average. Half or more of the survey participants chose the same importance for 10 out of 42 qualifications: "Positive ("can-do") attitude", "Debugging skills", "Inquisitive or Curious", "Strong Initiative (Self-Directed)", "Receptive to feedback (and code reviews)", "Learns things quickly", "Ability to work as an effective member of a team", "Big-picture thinking", "Ability to apply Software Engineering Best Practices", and "Can politely give feedback". Note that eight of these are professional skills, and not technologies or qualifications specific to software engineering.

With the highest average importance rating, *Positive ("can-do") attitude* stands out with the highest percentage of survey participants (67%) agreeing on its importance level. We speculate this may be due to the rapidly changing nature of software technology

and the need to continuously learn new things, but it could be due to corporate culture or other factors outside the scope of this study. The consistent emphasis on professional skills in the survey data might also be an indication of a lack of these qualifications in the general population of software engineers, or it is also possible these might be fundamental to understanding how software systems work. In future work we plan to identify the cause for this emphasis.

To explore whether or not there is variance across sub-fields ("functional areas") within software engineering, we can examine subsets of the survey responses into functional areas corresponding to each initial hiring manager's area. Will the engineers surveyed agree with similarly-focused managers about the most important qualifications?

Cloud Software. Cloud software engineers largely agreed with a cloud hiring manager about which qualifications are important. One of the interview subjects (a hiring manager) and many of the survey participants identified themselves as working on cloud software.

Seven of the hiring manager's top ten qualifications were also rated as important (4 or 5) by the majority of same-area survey responses. This indicates that those seven responses were very important to both the hiring manager and engineers working in the manager's area. Two of the three remaining top-ten qualifications were commonly rated "Neutral" (3) by interview subjects (see Table 1.

The survey participants' lowest-rated qualification (38% considering it a 2 of 5) was "Theory (i.e., computability theory)", which the hiring manager rated fourth most important (of 42). There is a wide variance in survey responses for this qualification, but the average rating was lower than all of the other qualifications for cloud software engineers. Future work should investigate why hiring managers and engineers disagree on the importance of theory.

14 (70%) of the 20 qualifications rated on average highest by cloud software survey participants are considered "professional skills" or personality attributes. This indicates a strong emphasis on professionalism and collaboration in the work environment.

"CS Core Fundamentals" was rated lower than 15 other qualifications, but had high concentrations of "3" and "5" ratings. Based on conversations during interviews with the hiring managers, the large number of "3" ratings may be due to a common assumption that all software engineers are screened for these fundamentals before they are hired.

DevOps. DevOps engineers also strongly agree with two DevOps hiring managers about top qualifications. Two interview subjects identified themselves as DevOps managers, and two survey participants indicated they worked on DevOps software. This is a small sample size and may not be fully representative, but there are indications of strong alignment between the managers and the engineers.

Seven of one and six of another manager's top ten qualifications were rated as "Important" or "Very Important" (4 or 5) by the survey responses. Of the remaining three qualifications for each manager, most were rated by survey participants as "Neutral" or "Very Important" (see Table 2).

The two hiring managers had different rankings, likely due to the differing needs of their teams. Both managers placed "Strong initiative", and "Teamwork" near the top of their rankings, but the remaining qualifications in their top-ten lists were different.

"Work Experience" and "Good on distributed teams" were consistently rated low by survey participants but high by the hiring managers. This disparity is likely biased by the hiring manager's specific needs, since the managers indicated their needs may be different than other groups and there are more than two DevOps groups at the subjects' company.

The DevOps survey participants were in alignment with each other for all qualifications except for three: "Growth Mindset", "Can Communicate via Online Tools", and "Exploratory (asks others what they are doing)", which suggests these attributes are important to the specific composition of a team and are not generalizable. Seven qualifications were rated "5" by both DevOps survey responses. Aside from "CS Fundamentals" (which has a wide variance in rating as mentioned earlier), all of these qualifications are also in the top third as rated by the entire survey population, suggesting wide agreement that these are important. Five of those seven qualifications were not computing-specific, and are instead professional skills or personality attributes.

"Strong Initiative" is near the top of the DevOps rankings but also near the top of the Cloud survey rankings. This is consistent with the average for *all* survey responses, where it is rated fourth-highest (53% of responses rated it a 5). That suggests this is a very important qualification across all sampled functional areas.

Firmware/Embedded and Other Areas of Software Engineering. The other two interview subjects managed a broad range of software engineering functions, which do not map well to the functions identified by survey participants as "Desktop" or "Firmware/Embedded" software. This data is more informative when used in combination with the other software functions (Cloud and DevOps) to analyze software engineering qualifications at this company broadly.

3.2.2 Comparison of Rankings. To identify how closely the two groups (hiring managers and software engineers) agree on importance of qualifications, we compared average rankings of the two groups.

The managers' ranking of qualifications is taken directly from the procedure followed during the interviews, described in Section 2. Ordering the qualifications by average survey response rating gives us a ranked list of qualifications from the software engineer perspective. We can identify how closely these rankings match by computing the average distance between positions on the two lists for each qualification.

On average, each qualification had a rank difference of 9 between ranking sets. "CS Core Fundamentals" and "Receptive to Feedback" changed rank the least (rank difference of 1). "Relationship-building" and "Ability to (nicely) ask others for help" moved the most with a difference in ranking of 26 and 27. This indicates inconsistency between the two rankings. This is not very significant due to the small survey sample, so it's not clear whether or not managers and engineers across functional areas agree on qualifications.

4 LIMITATIONS AND FUTURE WORK

This study gathered data from only one company, albeit a large company. Software engineering exists broadly in many sectors

Rank (mgr)	Qualification	Rating				
		1	2	3	4	5
1	CS Core Fundamentals	8%	0%	38%	8%	**46%**
2	Strong Initiative (Self-Directed)	0%	0%	22%	22%	**56%**
3	Willing to step a little out of their comfort zone	0%	11%	0%	**44%**	**44%**
4	Theory (i.e., computability theory)	15%	**38%**	23%	8%	15%
5	Problem-solving technique	0%	8%	23%	23%	**46%**
6	Sampling of programming/tech languages	0%	8%	**54%**	31%	8%
7	Learns things quickly	0%	22%	0%	22%	**56%**
8	Ability to work as an effective member of a team	0%	0%	8%	**69%**	23%
9	Algorithms	0%	15%	0%	**62%**	23%
10	Conceptual-level understanding of service-oriented architectures	0%	8%	**46%**	38%	8%

Table 1: Top qualifications ranked by a cloud software hiring manager and corresponding ratings from engineers. Values are percent of survey responses from cloud-focused participants who chose each rating (n=13).

Rank (mgr 1)	Qualification	Ratings				
		1	2	3	4	5
1	CS Core Fundamentals	-	-	-	-	**2**
2	Strong Initiative (Self-Directed)	-	-	-	-	**2**
3	Ability to work as an effective member of a team	-	-	**1**	-	**1**
4	Algorithms	-	-	-	**1**	**1**
5	Conceptual-level understanding of service-oriented architectures	-	-	-	**1**	**1**
6	Attention to detail	-	-	-	**1**	**1**
7	Work Experience	-	**1**	**1**	-	-
8	Positive ("can-do") attitude	-	-	-	-	**2**
9	Committed/follow0through (delivers what's promised)	-	-	**1**	-	**1**
10	Learns things quickly	-	-	-	**1**	**1**

Rank (mgr 2)	Qualification	Ratings				
		1	2	3	4	5
1	Good on distributed teams	-	**1**	**1**	-	-
2	Strong Initiative (Self-Directed)	-	-	-	-	**2**
3	Conceptual-level understanding of CI/CD platforms	-	-	-	**1**	**1**
4	Containers	-	-	**1**	**1**	-
5	Debugging Skills	-	-	-	-	**2**
6	Problem Solving Technique	-	-	-	**1**	**1**
7	"Growth Mindset"	-	**1**	-	-	**1**
8	Receptive to feedback (and code review)	-	-	-	-	**2**
9	Ability to work as an effective member of a team	-	-	**1**	-	**1**
10	Multi-task ability (and knowledge of whether or not they can do more or are overloaded)	-	-	-	**1**	**1**

Table 2: Top qualifications as ranked by two DevOps hiring managers and corresponding rating counts from engineers (n=2).

of the economy, and understanding desired qualifications more broadly could illuminate trends across the industry.

Especially interesting would be identifying why employers seem so focused on professional skills. Is this a symptom of educational gaps? Possibly this emphasis exists because universities are doing very well in technical education, so employers are looking for early indications of long-term career success. Understanding this emphasis would benefit from comparing software engineering desired qualifications to other, more mature engineering disciplines.

Although we reached out to more than 1000 engineers for our survey, the response rate was unfortunately very low (2.2%). We suspect this is because employees at this company have been subject to many surveys in the past year and are tired of volunteering their time to answer questions. Reattempting the same survey with participation incentives or at a less survey-saturated time of year may yield a bigger sample and provide more conclusive results.

Understanding top qualifications is only one step in Lundberg et al.'s recommended collaboration between industry and university [5]. Acting on this data requires comparing desired qualifications to accreditation criteria (such as ABET's [2]) and student/program outcomes at various universities.

5 CONCLUSION

Through interviews of hiring managers and a survey of software engineers, we've uncovered the most-desired qualifications for entry-level software engineers at one large Fortune 100 employer. Professional skills (teamwork, for example) top the lists of most important qualifications from both managers and engineers, and at least initially it appears the managers and engineers mostly agree on what is important. While the results are not strongly significant, they confirm findings from prior work that there is a general trend of demand for professional skill training over specific programming languages or technologies.

We now know more about what managers and engineering teams want for qualifications when seeking entry-level peers/hires at one large tech company. The focus on professional skills indicates an opportunity for many university programs to emphasize these more in their programs to help differentiate their graduates. If you manage a Software Engineering (or Computer Science) program at your university and something in this list is missing from your program's educational outcomes, perhaps these are topics to consider to increase your students' job placement.

Attribute	Category	1	2	3	4	5	Distribution of Ratings
Positive ("can-do") attitude	Attributes	0%	0%	13%	20%	67%	
Debugging Skills	Tech Skills	0%	0%	4%	50%	46%	
Inquisitive or Curious	Attributes	0%	0%	20%	20%	60%	
Strong Initiative (Self-Directed)	Prof Skills	0%	0%	20%	27%	53%	
Ability to (nicely) ask others for help	Prof Skills	0%	0%	12%	42%	46%	
Receptive to feedback (and code reviews)	Prof Skills	0%	0%	17%	33%	50%	
Committed with Follow-through (delivers what's promised)	Prof Skills	0%	7%	7%	40%	47%	
Attention to Detail	Attributes	0%	0%	13%	47%	40%	
Learns things quickly	Attributes	0%	13%	7%	20%	60%	
Problem-solving technique	Prof Skills	0%	4%	12%	38%	46%	
Ability to work as an effective member of a team	Prof Skills	0%	0%	8%	58%	33%	
Big-picture thinking	Attributes	0%	13%	13%	20%	53%	
Growth Mindset (knows their work may end up bigger than expected)	Attributes	0%	7%	13%	40%	40%	
Willing to step a little out of their comfort zone	Attributes	0%	7%	13%	40%	40%	
CS Core Fundamentals	Tech Skills	4%	0%	33%	17%	46%	
Relationship-building (cross-team)	Prof Skills	0%	12%	17%	38%	33%	
Algorithms	Tech Skills	0%	8%	25%	42%	25%	
Ability to apply Software Engineering Best Practices (e.g., Test-Driven Development, etc)	Tech Skills	0%	0%	33%	50%	17%	
Programming Languages	Tech Skills	0%	8%	25%	46%	21%	
Security Best Practices	Tech Skills	0%	8%	29%	38%	25%	
Documentation Skills (including self-documenting code, technical documents, diagrams, etc)	Prof Skills	0%	17%	21%	38%	25%	
Conceptual-level understanding of entire software lifecycle (requirements, construction, deploying, maintenance/updates, etc)	Tech Skills	0%	17%	21%	38%	25%	
Can communicate via online tools (e.g., Trello, GitHub Issues, MS Teams, etc)	Prof Skills	4%	4%	46%	12%	33%	
Exploratory (asks others what they are doing)	Attributes	7%	7%	33%	20%	33%	
Can politely give feedback (and review code)	Prof Skills	0%	8%	29%	54%	8%	
Ability to apply abstraction in design or implementation	Prof Skills	0%	12%	38%	29%	21%	
Ability to multi-task (and knowledge of when they are overloaded)	Prof Skills	4%	12%	29%	29%	25%	
Software Architectures	Tech Skills	0%	21%	21%	42%	17%	
Sampling of programming/tech languages	Tech Skills	0%	12%	38%	38%	12%	
Willing to help train others on new things (share their skills)	Prof Skills	4%	12%	33%	33%	17%	
Effective on distributed teams	Prof Skills	4%	17%	33%	25%	21%	
Conceptual-level understanding of CI/CD platforms	Tech Skills	0%	12%	46%	29%	12%	
General understanding of databases and their use (hadoop, mogodb, sql, nosql, etc)	Tech Skills	0%	21%	38%	29%	12%	
Ability to serve as different roles on a team	Prof Skills	0%	21%	46%	17%	17%	
Speaking/presenting skills	Prof Skills	4%	12%	46%	25%	12%	
Operating Systems	Tech Skills	4%	17%	33%	38%	8%	
Conceptual-level understanding of service-oriented architectures	Tech Skills	0%	25%	38%	29%	8%	
Cloud development techniques	Tech Skills	12%	12%	46%	17%	12%	
Understands economic/performance factors of their work (direct costs of engineering choices, e.g., AWS cost)	Prof Skills	4%	29%	46%	12%	8%	
Containers	Tech Skills	12%	21%	38%	25%	4%	
Theory (i.e., computability theory)	Tech Skills	12%	38%	29%	8%	12%	
Work Experience (internship or otherwise)	Prof Skills	25%	29%	25%	8%	12%	

Table 3: Importance ratings for qualifications as rated by survey respondents. "Distribution of Ratings" shows range, upper/lower quartiles, and mean rating (not median as standard in boxplots). Top-rated qualifications are consistently highly rated and those rated lower on average have more variance in importance to respondents.

REFERENCES

[1] Andrew Begel and Nachi Nagappan. 2008. Pair programming: what's in it for me?. In *ESEM '08: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (esem '08: proceedings of the second acm-ieee international symposium on empirical software engineering and measurement ed.). ACM, 120–128. https://www.microsoft.com/en-us/research/publication/pair-programming-whats-in-it-for-me/

[2] Engineering Accreditation Commission. 1996. *ABET Engineering Criteria 2000.* Technical Report. Accrediting Board for Engineering and Technology, 111 Market Place, Suite 1050, Baltimore, MD 21202.

[3] Michael Hewner and Mark Guzdial. 2010. What Game Developers Look for in a New Graduate: Interviews and Surveys at One Game Company. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10).* Association for Computing Machinery, New York, NY, USA, 275–279. https://doi.org/10.1145/1734263.1734359

[4] Paul Luo Li, Amy J. Ko, and Andrew Begel. 2020. What distinguishes great software engineers? *Empirical Software Engineering* 25 (1 2020), 322–352. Issue 1. https://doi.org/10.1007/s10664-019-09773-y

[5] Gunhild M. Lundberg, Birgit R. Krogstie, and John Krogstie. 2020. Becoming Fully Operational: Employability and the Need for Training of Computer Science Graduates. In *Proceedings of 2020 IEEE Global Engineering Education Conference (EDUCON).* IEEE, 644–651. https://doi.org/10.1109/EDUCON45650.2020.9125188

[6] Ashwin Mohan, Dominike Merle, Christa Jackson, John Lannin, and Satish S. Nair. 2010. Professional skills in the engineering curriculum. *IEEE Transactions on Education* 53 (11 2010), 562–571. Issue 4. https://doi.org/10.1109/TE.2009.2033041

[7] Alex Radermacher and Gursimran Walia. 2013. Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13* (New York, New York, USA). ACM Press, 525. https://doi.org/10.1145/2445196.2445351

[8] Larry Shuman, M. Besterfield-Sacre, and JACK MCGOURTY. 2005. The ABET 'professional skills'—Can they be taught? Can they be assessed? *Journal of Engineering Education* 94 (01 2005). https://doi.org/10.1002/j.2168-9830.2005.tb00828.x

[9] Justyna Szynkiewicz, Gunhild M. Lundberg, and Mats Daniels. 2020. Professional Competencies in Computing Education: Are They Important?. In *2020 IEEE Frontiers in Education Conference (FIE)*, Vol. 2020-October. IEEE, 1–5. https://doi.org/10.1109/FIE44824.2020.9273987

[10] Sander Valstar. 2019. Closing the Academia-Industry Gap in Undergraduate CS. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19).* Association for Computing Machinery, New York, NY, USA, 357–358. https://doi.org/10.1145/3291279.3339440

[11] Sander Valstar, Sophia Krause-Levy, Alexandra Macedo, William G. Griswold, and Leo Porter. 2020. Faculty Views on the Goals of an Undergraduate CS Education and the Academia-Industry Gap. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20).*

[12] Xin Xia, Zhiyuan Wan, Pavneet Singh Kochhar, and David Lo. 2019. How Practitioners Perceive Coding Proficiency. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE).* IEEE, 924–935. https://doi.org/10.1109/ICSE.2019.00098