

EMPLOYERS' NEEDS FOR COMPUTER SCIENCE, INFORMATION TECHNOLOGY AND SOFTWARE ENGINEERING SKILLS AMONG NEW GRADUATES

Christopher Scaffidi

School of Electrical Engineering and Computer Science, Oregon State University, US

ABSTRACT

Graduates of computer science programs often lack skills that employers desire among software developers. These include, for example, weaknesses in the areas of collaboration, communication, and software testing. Further research can help to refine this list by providing insight into additional skills that are of rising or regional importance. This paper therefore presents a study aimed at uncovering desirable technical and soft skills for graduates of computer science in the Pacific Northwest region of the United States. Interviews of 11 employers, including both managers and recruiters, highlighted the prominent importance of skills related to web development, relational databases, and testing. Additionally, it spotlighted not only widely-recognized soft skills such as those related to collaboration and communication, but additionally on skills tied to personal attributes such as innovating, coping with ambiguity and learning quickly. The results provide insights for what skills and personal attributes to include in a future survey of employers aimed at quantifying the importance of skills on this list.

KEYWORDS

Computer science education, Software engineering education, Information technology education

1. INTRODUCTION

Immediately after graduation, students in computer science typically struggle with becoming productive as software developers. Common difficulties reported by novice professionals include weaknesses in soft skills such as collaboration and communication, as well as in technical skills related to source code control and testing [1]. Direct observation of developers has confirmed that these skills play a large role in daily work and present substantial challenges [2]. Surveys further have revealed that developers learn many such skills on the job, reflecting gaps between skill upon graduation and respective accumulated skill thereafter [3, 4].

What developers perceive about their strengths and weaknesses may differ from those perceived by other stakeholders, such as the educators who train them. For example, one study of the software industry in Philippines revealed a significant disconnect between the perceptions of workers about their weak human relations and collaboration skills (particularly leadership), versus perceptions of educators [5].

Studies of employers' perceptions of recent graduates' skills can help reveal weaknesses that potentially impact worker productivity. One literature review of empirical studies since 1995 revealed many such gaps between industry expectations and the abilities of graduates, the most significant of which were related to communication, teamwork, problem solving, testing and

programming [6]. Exploring specific papers from that period provides more detailed insight into topical gaps.

Several recurring themes appear among the literature cited above, the most prominent of which appears to be collaboration and communication, and another of which is software testing. Universities can train future software developers in such skills within the context of courses on computer science, software engineering, and information technology. Hence, ongoing research can inform the development of university curricula in two ways.

First, studies can aid in updating the list of specific topics requiring attention. For instance, a paper based on a 1998 survey included a list of 25 “least important” topics for academia to focus on further emphasizing, two of which were artificial intelligence and robotics [7]. In the nearly two decades since the survey, both areas of expertise have grown in importance, leaping to the forefront of technological innovation and commercial development, with potentially transformative effects on the culture and economy [8]. Likewise, the papers cited above have little or no mention of big data, cloud computing, mobile app development or other new solution domains that may have become important to employers. Skill lists need continual refreshing based on fresh empirical studies.

Second, additional studies can reveal regional differences that may inform the enhancement of curricula. Three examples of potentially region-specific insights were (1) that a study of Australian employers noted a particular need for workers using information and communication technology to have more business acumen [9], (2) a study of North Dakota State University’s employer contacts highlighted the potential benefit of giving students more project experience prior to graduation and employment as software developers [10], and (3) a study of knowledge gaps in the United Kingdom identified web-based programming as the top area of concern for software developers [11]. Comparing the needs to employers can then suggest opportunities related to creation, expansion and sequencing of courses in academic programs [12, 13].

This paper therefore presents a study aimed at uncovering desirable technical and soft skills for graduates of computer science from Oregon State University. As in one of the regional studies cited above [10], the study used interviews to generate a list of such skills, suitable for inclusion in a future survey aimed at quantifying the importance of skills on this list.

2. MATERIALS AND METHODS

The study’s materials and methods were chosen to address three specific research questions:

- **RQ1:** What technical skills do employers value of software developers?
- **RQ2:** What soft skills do employers value of software developers?
- **RQ3:** To what extent do different employers value different skills?

2.1. Data collection

Participants were recruited by sending emails to a list of employers who had indicated interest in contributing feedback about the relationship between software engineering education and their hiring needs. Approximately half of these had indicated this interest while previously participating in on-campus recruiting events; the other half had indicated interest via LinkedIn.

When a potential participant replied to the email, expressing continued interest and a desire to be interviewed for the current research, a copy of the consent form was sent to explain the details of the study. They could then indicate a time to be interviewed via phone or Skype.

The interview consisted of seven primary questions (Table 1), potentially with follow-up questions to clarify answers. Each interview generally required between 10 and 15 minutes. During the phone call, notes were manually written. The calls themselves were not recorded, nor transcripts created.

Table 1. Questions asked during the semi-structured interview.

Item	Question text
1. Respondent Role	Please describe your role or job in your organization.
2. Previous Hires	Has your organization previously hired students from the computer science program at Oregon State University?
3. Future Hires	In the future, what roles or jobs might you anticipate hiring graduates of our programs for?
4. Technical Skills	What kinds of technical skills or knowledge will be important for our graduates to have in order to fill those jobs?
5. Soft Skills	What other qualities, such as soft skills or other experiences, will be important?
6. University Evaluation	What do you see as the strengths and weaknesses of our university programs, relative to the skills that you're looking for in our graduates?
7. Other Feedback	Finally, what other feedback do you have about potential improvements for our programs?

2.2. Data analysis

Interview notes were analysed using latent (semantic) thematic analysis, a technique common in other training disciplines such as nursing [14]. This method involves several steps: (1) identify general themes, (2) develop a specific set of codes, (3) try applying the code set to the data, (4) create a list of changes to the code set, (5) iterate the preceding two steps until no additional changes to the code set remain apparent, and (6) name the themes identified with the codes. Completing these steps typically requires multiple passes through the data. Each time, in step (4), necessary changes to the code set become more apparent if instances of a code are collected together to identify how they are similar or different, and if areas of data that have no associated code are identified and reviewed for new codes.

The codes generated for this study were oriented around technical and soft skills. Specific programming tools were not coded primarily because of this focus on skills rather than tools. An additional for not reporting names of specific programming tools is that it appeared every participant's company used a unique set of tools, so reporting this information for a participant could potentially serve to identify organizations and thereby compromise confidentiality.

The thematic analysis provided information suitable for addressing the three research questions. For RQ1 and RQ2, themes were associated with either technical or soft skills, respectively. Note that technical skills may have been mentioned by a given participant in response to any question of the interview (not necessarily just question 4 of Table 1—it was quite often the case that interviewees came back to topics later in the interview); likewise, soft skills could be mentioned at any point in the interview. Finally, RQ3 was answered by reviewing the cross-tabulation of skills mentioned versus participants, to assess whether any obvious patterns emerged in terms of who referenced each skill.

3. RESULTS

A total of 11 people participated in interviews. Of these, 3 were practicing developers or analysts, 3 managed developers at the team or business level, 3 were internal corporate recruiters, and 3 were external recruiters (1 of which also was a practicing developer). Analysing notes from interviews yielded a total of 16 themes each reflected in at least 2 participants' answers (plus one theme, user interface design, discussed in detail below). Of these themes, 11 involved technical skills and 5 involved soft skills. Table 2 summarizes responses.

Table 2: Cross-tabulation of themes mentioned by participants.

Participant:	1	2	3	4	5	6	7	8	9	10	11	(Total)
Technical/domain skills												
Agile methods				X		X						2
Business & entrepreneurship		X		X								2
Embedded development				X	X	X						3
Mobile development							X	X		X		3
Non-relational data/big data	X	X	X				X			X		5
Relational databases	X		X		X	X	X	X				6
Source code management	X		X							X		3
System administration			X		X	X				X		4
Testing and quality assurance				X	X	X		X		X		5
User interface design				X								1
Web development	X	X	X			X	X	X	X	X	X	9
Soft skills / personal attributes												
Ability to collaborate	X	X		X	X	X			X	X		7
Ability to communicate well		X	X	X	X	X			X	X	X	8
Ability to cope with ambiguity	X	X		X				X				4
Learning and curiosity		X	X		X			X	X	X	X	7
Passion / drive to innovate			X		X			X		X	X	5

3.1. Technical skills

Agile methods

Two interviewees mentioned agile methods. Participant #4 stated, "Waterfall is evil" and then described how their software developers engage entirely in agile methods. They expected new hires likewise to have relevant experience and be ready to participate as full members of the team. Participant #6 elaborated that the company used Scrum, in particular.

Business & entrepreneurship

Two participants indicated they valued new hires who had an aptitude and interest for business and entrepreneurship. Participant #2 noted applicants have widely varying levels of business and background and attitude toward entrepreneurship. Participant #4 precisely characterized several aspects of business skill that the organization valued. These included “simple things like creating a budget” as well as the ability to think about the net present value of a proposed software project or feature. The participant conceded as a student some time ago, “I would have run the other way [from business courses], but now it’s my world.” He also noted the value of basic statistics skills in the context of business-oriented thinking by software developers.

Embedded development

Three interviewees indicated a desire to hire graduates with skills in embedded development. Participant #4 described how their business involves selling hardware with embedded firmware, which calls for hiring workers with the skills to create that software. Participant #6 mentioned firmware development in passing. Participant #5 noted that at that company, software engineers frequently must coordinate system development efforts with work by mechanical and electrical engineers who have primary responsibility for hardware development, and that having hybrid skills (software plus some hardware development ability) is best.

Mobile development

Three interviewees mentioned mobile development. Participant #7 (a recruiter) indicated the desirability of skills in platform-specific (Android/iOS) and cross-platform programming environments, further indicating that “25-40% of mobile projects” involved cross-platform environments. Participant #8 said there was only a small need for mobile skills at that company, and Participant #10 (a recruiter) said that although mobile is “out there,” it is nonetheless only a small hiring target at present.

Non-relational data/big data

Five participants mentioned needing to hire software developers who had skills in dealing with non-relational data and/or big data. Participant #1 clarified that this often involves writing code to process text data into a usable (often relational) form, while Participants #2, #3 and #7 mentioned specific big data tools as well as the ability to apply statistics. Participant #10 noted specific programming languages and platforms related to big data and indicated that when they find people with relevant skills, sometimes “companies create positions just to hire someone.” Participant #7 said that hiring data scientists was extremely challenging right now, and that this was often a distinct job from software development.

Relational databases

One of the most prevalent themes, relational databases were brought up by six participants. Participant #1 noted the difficulty of finding recent students with adequate experience in this area. Participants #3, #5 and #6 mentioned in passing specific database systems or database management in general. Participant #7 described how an interviewing test wherein applicants had three hours to implement an application that would store information in a database. Participant #8 explained the need for workers with an ability to craft SQL with an attention to performance.

Source code management

Three interviewees showed an interest in software developers with experience in source code management tools and/or practices. Participant #1 said that company especially valued software developers who could participate in joint management of source code while not being protective or “territorial” about code. Participant #3 mentioned a specific source code management platform. Participant #10 said that company often examines applicants’ online code repositories during the hiring process.

System administration

Four participants noted aspects of system administration skills that benefit software developers. Participant #5 characterized these as being related to corporate information systems (IS), database management, and network security. Participant #6 noted the need for expertise in enterprise system management. Participant #10 mentioned devops, which encompasses both software development and the operation of information technology (IT) systems, and emphasized that company's focus on hiring people who came from a development background toward devops rather than an IS background. Participant #3 indicated needing to hire a few people to support servers shared by different projects.

Testing and quality assurance

One of the other most oft-mentioned areas of technical skill, testing and quality assurance more broadly, was mentioned by five interviewees. Participant #4 said workers frequently had to engage in testbed engineering, which involved creating real and virtual machines, to perform automated testing with motors, controllers and other hardware in the loop; these responsibilities, furthermore, involved data collection and analysis. Participant #5 mentioned the need for skills in validation of requirements and verification of systems. Participant #6 indicated that applicants frequently fail to be hired due to poor testing skills. Participant #8 said that quality assurance skills should include experience with unit testing and integration testing.

User interface design

Of the emergent themes, only that of user interface design was somewhat unclear based on interview notes. More precisely, it was explicitly mentioned only by Participant #4, and only in passing with no additional details. However, Participants #7 and #9 did mention one framework usually used for implementing user interfaces (with no mention of whether they valued relevant design skills), potentially implying the relevance of this theme to three interviewees.

Web development

By far, web development was mentioned by the most participants—a total of nine out of eleven. Participants #1, #3, #6 indicated that web development was the primary hiring need. Participants #7 and #8 indicated it was one of the primary hiring needs (alongside those of desktop development, mobile development and/or big data science). Participant #2 indicated a desire to find software developers with the ability to design web applications for diverse populations of users. Participant #3 needed to hire developers skilled in web-based generation and delivery of reports. Participants #10 and #11 tied web development to cloud development. Most of these participants gave the names of one or more languages, tools and platforms used internally for web development. Some, such as Participant #3 and #7, emphasized the importance of using tools and design patterns in academic training that matched those used in industry; others, such as Participants #1 and #9, said that applicants' specific language and platform experience mattered relatively little because it was relatively easy for developers experienced with one to pick up another.

3.2. Soft skills and related personal qualities

Ability to collaborate

Seven interviewees indicated needing workers who could collaborate with teams. Of these, two also indicated a focus specifically on teams that applied agile methods (Section 3.1). Although Participant #10 only noted collaboration in passing, most other interviewees gave details about specific aspects of this soft skill that they valued. Participant #1 said that teamwork frequently involved doing code reviews with one another. Participant #2 emphasized the importance of a worker being able to see how one's work fit into the "bigger picture" of the system under development. Participant #4 discussed the importance of being able to negotiate scope, schedule and resources, as well as the ability to "manage upward" (pushing back on supervisors and other

superiors in constructive ways). This participant further explained, “We do not value people sitting at their desk and writing code all day” and reiterated the need for software developers to coordinate their work with other stakeholders including people who are not directly involved in system creation. Participant #5 noted that company was appreciative and respectful when interns accepted fault in the context of team work. Participant #6 emphasized the need for leadership and team-building skills and that applicants should must articulate during interviews how they demonstrated leadership, teamwork and initiative during past projects. Participant #9 said that company went so far as to construct group design exercises, during the interview process, wherein they observed applicants interact with one another in the context of assigned design problems.

Ability to communicate well

A total of eight participants emphasized the importance of communication skills. Participant #2 mentioned it in passing. Participant #3 noted the need to communicate with other members of a global team. Participant #4 mentioned the need to articulate and document requirements. Participant #5 explained that being straightforward and honest in communication was essential. Participant #9 said that the “number one weakness” of applicants was weak communication, especially verbal communication during interviews; the interviewee also commented that visual presentation in terms of clothing worn also served (and often failed to serve) as a crucial element of tacit communication during interviews. Participant #10 valued whether workers could explain technology in non-technical terms, ask insightful questions, and be personable when performing client-facing roles. Participants #6, #9 and #11 emphasized the importance for applicants to tell stories about past projects in ways that demonstrated knowledge and expertise.

Ability to cope with ambiguity

Four interviewees mentioned ways in which they valued software developers who could cope with different forms of ambiguity. Participant #2 said that developers often had to do quite a bit of data cleaning, especially when processing textual data and data from legacy databases, and that much of this data was “messy.” Participant #4 said that projects often involved a good deal of “real-world chaos” that workers resolved through collaboration. Participant #8 noted that it was rare to find a recent graduate who does not view the job as one of simply implementing “requirements as given,” emphasizing that a key differentiator is whether a worker can take a relatively vague directive and “run with it.” The same participant further explained a desire to find people capable of viewing difficult problems from multiple viewpoints.

Learning and curiosity

One of the most extensively discussed and often-mentioned themes was whether software developers—and, particularly, recent graduates—had an innate intellectual curiosity and desire to learn. The seven who mentioned this included Participants #3, #5 and #10, who gave no additional detail. Participant #2 characterized this in terms of being “lifelong learners” and noted that applicants have widely varying level of hunger to learn. Participant #8 describe it as having a high curiosity pushing them beyond their own individual responsibilities, and a desire to build expertise. Participant #9, whose company provides classes internally to their software developers, emphasized wanting employees who could learn fast and stated a key consideration during hiring was the question, “Can I make an Enterprise Architect out of you?” (within a 10-year time horizon). Participant #10 characterize the “drive” for learning in desirable software developers with the “I don’t care” attitude evidenced by many recent graduates.

Passion / drive to innovate

Five participants mentioned seeking workers who had some sort of passion or desire to innovate. Participant #3 said that company wanted software developers who were passionate and eager to be involved in many projects, who had personal projects of their own, and who would go “above and beyond” the basic requirements of a given project. Participant #5 emphasized the need for

people who could demonstrated ingenuity, creativity, and a desire to do research and development even in the context of day-to-day operations to help “evolve” the way that company did design and production of systems. Participant #8 mentioned watching for whether recent graduates were “excited and interested” about “thinking beyond homework assignments,” a sentiment that Participant #11 also expressed. Participant #10 framed this issue in terms of whether recent students were interested in going beyond “projects that are given” toward “solving [problems] on their own” with little guidance about where to focus effort.

4. DISCUSSION

The results address the research questions that motivated this work.

RQ1: What technical skills do employers value of software developers?

Interviewees’ answers revealed the need for 11 different technical skills. These included 5 technical skill areas related to a solution domain (Embedded development, Mobile development, Non-relational data/big data, Relational databases, and Web development), plus 6 skills related to methods and practices that apply in most solution domains (Agile methods, Business & entrepreneurship, Source code management, System administration, Testing/quality assurance, and User interface design). The most prevalent skills were those related to web development and relational databases. Some of the technical skills that currently get a large amount of attention in the popular press—such as those related to mobile development and big data—were not mentioned by many employers. Whereas mobile development was downplayed in importance by two interviewees, the few who mentioned big data emphasized the difficulty of hiring enough people with relevant skills. Thus, the explanation for why some are rarely mentioned may vary, either because a skill is one that employers are not yet interested in pursuing at present or because they are interested in pursuing but not yet able to access. One area of skill, user interface design, deserves additional study because it was difficult in some cases to tease out the extent to which employers implicitly sought those skills.

RQ2: What soft skills do employers value of software developers?

Employers valued 5 different soft skills. These included collaboration and communication, which have often appeared in other studies of soft skills needed by workers in many occupations, including software development (Section 1). Interviews also revealed 3 additional areas of strength that in some sense border on skill but that in another sense represent manifestations of deeper personal attributes. The ability to cope with ambiguity appeared in both skill with handling messy data, messy processes, and messy requirements; it is unclear to what extent this skill, perhaps tied to a deeper sense of comfort amid mess, can be taught. Likewise, ability to “learn fast,” as one participant put it, may be a skill, but it also is related to an innate curiosity and an absence of what another participant characterized as an “I don’t care” attitude. Finally, similarly, the ability to innovate manifested primarily in workers going beyond stated requirements and pushing to find creative solutions—again, perhaps reflecting a deeper characteristic worthy of further research.

RQ3: To what extent do different employers value different skills?

Interviewees demonstrated substantial diversity in their answers. No two participants mentioned precisely the same sets of skills. In fact, no two mentioned the same set of solution domain skills (see RQ1, above), nor the same set of domain-independent skills, nor the same set of soft skills. Employers also differed widely in terms of the extent to which they emphasized technical skills (e.g., Participants #7 and #6), soft skills (e.g., Participants #9 and #11) or a balanced mix. In a few cases, one skill subsumed another:

- Everyone who mentioned non-relational data also mentioned web development
- Everyone who mentioned mobile development mentioned web development

- Everyone who mentioned agile methods mentioned collaboration
- Nearly everyone who mentioned collaboration also mentioned communication (and vice versa)
- Everyone who mentioned passionate innovation also mentioned learning/curiosity
- In addition, technical skills mentioned by few interviewees (e.g., Agile methods and User interface design) could be subsumed into other technical skills.

These relationships indicate that, despite the wide diversity in employer needs, opportunities nonetheless may exist for partitioning and integrating skills with one another within academic courses or tracks. For instance, because non-relational data did not appear except within web development, training in non-relational data could be integrated within (or taught as an expansion upon) training in web development. Likewise, the tight coupling of agile, collaboration and communication suggests an opportunity to teach all three within a specific course focused on the integrated practice of these skills.

5. RELATED WORK

Other studies have also examined the mismatch between university curricula and the needs of practicing software developers. They have done so with varying methods and categories of participants.

One group of studies has primarily involved interviewing or otherwise surveying software developers and attempting to uncover what skills they had to gain through on-the-job training, due to inadequacies in their preparation upon graduation. Some of this work is quite out of date but potentially relevant. For example, one widely-cited 1998 survey of employers led to papers arguing that the most crucial topics needing better academic coverage were related to programming languages, data structures and software design [4][7]. A more up-to-date study from 2014 involved interviewing 9 software developers about the training gap, which generated insights for a subsequent survey [3]. The most often-mentioned needs for better training were in the areas of working with source code control, maintaining code, and testing practices. This study also strongly emphasized the importance of workers being capable of teaching one's self from examples and of using peers as a resource, particularly in the context of team projects. A study involving interviews of managers confirmed the need for better training of students in use of tools, including those for source code control and testing; it also emphasized the importance of communication skills [10]. Overall, the current survey confirms the ongoing need for training students in these areas, and it expands the list further to include many other important technical and soft skills (Table 1) not mentioned much in earlier work, such as mobile development, non-relational/big data, and web development.

Other work has followed an observational, rather than interview-based, methodology. For example, one study observed 8 recent graduates after they joined Microsoft Corporation as software developers, which revealed numerous on-the-job challenges in the transition from school to practice [1]. These included problems with coordinating work with colleagues and knowing when to ask for help, consistent with the current study's emphasis on soft skills as an important trait sought by employers. Moreover, these new employees often struggled with using source code control and related tools, again consistent with the current study. Further analysis by a second study of the data from the same 8 developers at Microsoft highlighted the important role of socialization and social structure in enabling new graduates to master the skills required for successful practice as a software engineer [2]. Participants explicitly stated that university curricula inadequately prepared them for collaborating in large teams beyond the 2-3 person teams typical in school. The current study complements observational studies such as these by

showing employers are not only aware of these deficiencies in new workers, but also that they actively seek to find new hires who possess the targeted strengths and skills.

A third method of studying this topic was that of the focus group, used in one study where 11 Australian employers of software developers gathered for breakfast and discussed their hiring needs [9]. Participants could write and stick Post-It notes onto the wall, then comment on one another's notes. From the thematic analysis emerged a list of skills sought by employers, with communication skills and teamwork at the top of the list. The study also notes the importance of "technical skills" in "IT aptitude," "relevant technical abilities" and "comp sci / programming skills [sic]," but offers little further information about this area of need. The current study's methodology eliminates the risk of "groupthink" inherent in the focus group method, and it expands the list of needed skills both in breadth and in level of detail.

Finally, some work has focused on measuring the *divergence* in views between educators and practicing professionals, surveying each and then comparing responses. A regional study in Philippines, for instance, found the biggest divergence in training for problem solving, which educators considered slightly more important than practitioners did; in other areas, however, there were no statistically significant differences [5]. Both groups, for example, placed a high degree of emphasis on training in information technology tools, critical thinking, communication, teamwork, and many of the other skills that appear in Table 1. The current study highlights the fact that these findings generalize beyond Philippines and apply to the context of software development companies in the US Pacific Northwest, as well.

6. LIMITATIONS AND THREATS TO VALIDITY

As with any research, this study exhibited certain limitations and threats to validity. First, interviews necessarily require introduction of a sample frame (the set of people from whom participants are recruited), which may not represent the entire population of a country or of the entire world. Second, sampling can introduce bias, in that it is possible that some members of the sample might be more likely to participate, which can alter conclusions if their answers systematically differ from those who do not participate. These are potentially problematic in the context of this study, as it involved only 11 participants (comparable to the sample size in interview-based studies cited in Section 5). Third, because thematic analysis in its most basic form omits peer checking [14], coding error may have resulted, a problem compounded, fourth, by coding from interview notes rather than raw interview audio.

Given these limitations and threats to validity, to what extent can this study's answers to its three research questions be trusted? The conservative response is to view them as provisional, essentially constituting hypotheses worthy of further exploration by subsequent research. For example, the analysis answering RQ1 can identify a set of technical skills that appear promising, which subsequent research can then investigate in more detail.

7. CONCLUSIONS

This study has highlighted the broad diversity of technical and soft skills sought by employers of software developers and revealed differences in the prevalence of, and relationships among, these skill needs. The results lay the foundation for future work in four areas.

First, further research could address methodological weaknesses of thematic analysis on interview notes and the relatively small sample size. The methodology employed is suitable for generating insights for future work, not for generating definitive conclusions. Further work focused on triangulation could help to refine and establish such conclusions. For example, a multiple-choice

survey of different employers could present a second viewpoint (thereby helping to limit the threat of sample bias) and eliminate the need for open coding (thereby negating the risk of coding error).

Second, these results—pending confirmation by further study, as noted above—suggest potential opportunities for expanding, refining and reinforcing computer science curricula. For example, schools that do not require all students to take courses in embedded and mobile development might take comfort from this study in that few employers sought workers skilled in that area, but those schools entirely lacking courses that emphasize testing and collaboration might wish to shore up those weaknesses. In addition, the fact that some skills appeared to subsume others (Section 4) might suggest insights for integration and/or sequencing of courses.

Third, the study revealed several aptitude-centric skills—curiosity/learning and passion/innovation—that have been mentioned yet not consistently been prominent in prior studies of what skills employers seek in software developers (Section 1). Additional study could assess if these are of rising importance or of primarily regional interest. In addition, leveraging existing research from the learning sciences and on training innovators could aid in developing methods for evaluating the extent to which computer science graduates possess or lack these qualities and associated skills.

Finally, employers noted that they often pay attention to prior project work (especially outside classes), and to how applicants spoke about past work, when attempting to judge applicants in terms of drive to learn and to innovate. Further work could investigate related issues such as what kinds of projects are most effective at building strength in these areas, as well as what forms of project presentation are most effective at communicating strengths (e.g., A working app? Source code?). Other questions are to what extent students are aware of the need for projects to build and demonstrate these strengths, to what extent increasing this awareness causes students to engage in such projects, and to what extent does this increase in engagement enhance employability? Answering such questions and dozens like them could aid in enhancing the academic training of future software developers and build long-term career success.

REFERENCES

- [1] Begel, A., and Simon, B. (2008) Struggles of new college graduates in their first software development job. *ACM SIGCSE Bulletin*, 226-230.
- [2] Begel, A., and Simon, B. (2008) Novice software developers, all over again. *International Workshop on Computing Education Research*, 3-14.
- [3] Exter, M. (2014) Comparing educational experiences and on-the-job needs of educational software designers. *ACM Technical Symposium on Computer Science Education*, 355-360.
- [4] Lethbridge, T. (2000) What knowledge is important to a software professional? *IEEE Computer*. 33(5), 44-50.
- [5] Bringula, R., Balcoba, A., and Basa, R. (2016) Employable skills of information technology graduates in the Philippines: Do industry practitioners and educators have the same view? *Western Canadian Conference on Computing Education*, 1-6.
- [6] Radermacher, A., and Walia, G. (2013) Gaps between industry expectations and the abilities of graduates. *ACM Technical Symposium on Computer Science Education*, 525-530.
- [7] Lethbridge, T. (2000) Priorities for the education and training of software engineers. *Journal of Systems and Software*. 53(1), 53-71.

- [8] Brynjolfsson, E., and McAfee, A. (2014) *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*, WW Norton & Company
- [9] Hamilton, M., Carbone, A., Gonsalvez, C., and Jollands, M. (2015) Breakfast with ICT employers: What do they want to see in our graduates? *Australasian Computing Education Conference*, 29-36.
- [10] Radermacher, A., Walia, G., and Knudson, D. (2015) Missed expectations: Where CS students fall short in the software industry. *CrossTalk: The Journal of Defense Software Engineering*. 28(1), 4-8.
- [11] Kitchenham, B., Budgen, D., Brereton, P., and Woodall, P. (2005) An investigation of software engineering curricula. *Journal of Systems and Software*. 74(3), 325-335.
- [12] Amiri, A., Banari, M., and Yousefnezhad, N. (2011) An investigation of undergraduate software engineering curriculum: Iranian universities case study. *IEEE International Conference on Computer Science & Education*, 638-644.
- [13] Hanna, S., Jaber, H., Jaber, F., Al Shalaby, T., and Almasalmeh, A. (2014) Enhancing the software engineering curriculums: A case study of the Jordanian universities. *IEEE Conference on Software Engineering Education and Training*, 84-93.
- [14] Vaismoradi, M., Turunen, H., and Bondas, T. (2013) Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & Health Sciences*. 15(3), 398-405.

AUTHOR

Christopher Scaffidi earned a Ph.D. in software engineering from Carnegie Mellon University and is currently an associate professor of computer science in the School of EECS at Oregon State University. His research interests are where human-computer interaction and software engineering intersect. Most of his current projects aim to help software users to create code for themselves, and to effectively share that code with one another.

