A Quantitative Study of Faculty Views on the Goals of an Undergraduate CS Program and Preparing Students for Industry

Sander Valstar, Caroline Sih, Sophia Krause-Levy, Leo Porter and William G. Griswold
University of California, San Diego
{avalstar,csih,skrausel,leporter,wgg}@ucsd.edu

Abstract

Although elements of the academia-industry gap have been studied extensively, these studies have mostly ignored the primary stakeholder for changing academia: faculty. Building on a recent qualitative study that revealed a wide range of faculty views on the gap, this study quantitatively examines faculty views through a survey on the goals of CS education, how CS programs should address the academia-industry gap, and which barriers prevent adoption of remedies. Analysis of the 249 responses reveals that a majority of faculty share common goals in supporting student preparation for a career in industry. Moreover, faculty strongly view their own institutions as the prime party responsible for student preparation for careers in both academia and industry. We also find that whereas faculty are generally in agreement on what could be improved to provide students with better industry preparation, some reported far greater barriers to implementing those improvements than others.

CCS Concepts

• **Social and professional topics → Computer science education**; **Software engineering education**.

Keywords

computer science education, academia-industry gap, faculty views, survey

ACM Reference Format:
Sander Valstar, Caroline Sih, Sophia Krause-Levy, Leo Porter and William G. Griswold. 2020. A Quantitative Study of Faculty Views on the Goals of an Undergraduate CS Program and Preparing Students for Industry. In *Proceedings of the 2020 International Computing Education Research Conference (ICER '20), August 10–12, 2020, Virtual Event, New Zealand.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3372782.3406277

1 Introduction

Computer Science students and their ultimate employers have long lamented that their undergraduate education did not adequately prepare them for their career. Many stakeholders in the so-called *academia-industry gap* have been studied over the years, including students [5, 16], recent graduates [4, 8], employers and practitioners [12, 27]. Given that faculty are essential to closing any such gap, it is surprising that the faculty view on this topic had remained relatively unstudied.

Recently, we conducted an in-depth interview study of 14 computer science faculty to learn the range of faculty views on the goals of a CS program, where they felt their programs were falling short, and what was hindering improvement [34]. One might think that faculty would claim "we are not a vocational school" in regards to questions about closing the academia-industry gap. Such views were indeed voiced by a small number of the interviewed faculty, but the other faculty desired to teach a curriculum relevant to industry preparation. These faculty cited a range of barriers to better preparing students for industry, including lack of knowledge of industry practices, lack of resources, and large class sizes.

Although this prior qualitative study [34] uncovered a wide range of faculty views on these matters, including on the academia-industry gap, the relative support for those views in the broader community of CS faculty is unknown. Is the attitude of "we are not a vocational school" a major contributor to the academia-industry gap, or is the larger contributor these barriers to improvement cited by faculty? To learn the answers to these questions, we sought to quantify faculty views on the goals of a CS education, how best to prepare students for their academic or industry career, and what barriers they perceive inhibiting achievement of those goals.

Building on the results of our prior qualitative study [34], we conducted a broad survey study to establish which views are predominate among faculty, and how those views vary by institutional type and other factors. Specifically, we translated the range of faculty views captured in our prior qualitative work into multiple choice questions, and then sent them out as a survey to thousands of CS faculty across the globe, using mailing lists, contact information from Google Scholar, and scraping e-mail addresses from computer science department web sites (described further in Section 3). In analyzing the survey responses, we found the following:

- In terms of the goals of a CS education, we found the highest support for traditional CS goals such as students learning problem-solving, programming, and how to acquire new computing knowledge on their own.
- We found consistently high support for imparting skills and experiences related to industry preparation. Right behind the above-cited traditional CS goals were understanding the real-world context of computing as well as soft skills, including time management and working in teams, ethics and the role of computing in society, and software engineering skills. Yet, we found split support for goals related to personal development, such as helping students explore new interests or discover their path in life. When asked about the relative importance of supporting industry preparation and academic (graduate school)

ICER '20, August 10–12, 2020, Virtual Event, New Zealand
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7092-9/20/08.
https://doi.org/10.1145/3372782.3406277

preparation, faculty favored industry preparation. Faculty also felt that much of the responsibility for industry (and academia) preparation should fall on the shoulders of their institution.

- The majority of faculty expressed support for improving their courses by integrating material and experiences related to industry preparation into their courses, such as software engineering best practices and soft skills. This varied widely by the primary course taught by faculty, with faculty who teach courses related to algorithms being more split about the appropriateness of preparing students for industry in their course, whereas faculty who teach courses involving the construction of software rate industry preparation more highly.
- Faculty were split about how many and which barriers they encounter to offering students better industry preparation, with roughly half of participants perceiving only a few barriers and a quarter of participants perceiving substantial barriers.

With generally strong faculty support for industry preparation, and the split views on the difficulty of providing that preparation, there remains a puzzle about why the gap remains.

In the next section we review the literature on the academia-industry gap. We then describe our methods and the survey results, and finally close with a discussion of the results and a conclusion.

2 Background

2.1 The Academia-Industry Gap

The gap between recent CS graduates' knowledge and industry expectations has been a well-known problem for decades, yet it still exists. Industry practitioners have stated that many recent graduates lack relevant technical and non-technical skills, leading to failed interviews and requiring graduates to go through large amounts of additional training upon being hired. Recent graduates are known to lack the ability to test software in real world settings [25], lack relevant project experience [26, 27], have poor communication and team building skills [4, 26, 27] and lack the ability to configure and use software tools [26].

A study by Radermacher and Walia, interviewed 23 industry professionals to understand areas where recent graduates were underprepared. They found the most common issue to be struggling with software tools, followed by misunderstanding the expectations of the job. Another study interviewed recent graduates and found that there were six main areas where the graduates fell short, these included: "communication, team work, working on large, long-lasting projects that are open-scoped and using complex software systems" [8].

To gain a deeper understanding of new graduates' experiences in industry, Begel and Simon followed eight new graduates during their first six months in industry. They found that the graduates had many misconceptions about their new roles such as "I must do everything myself so that I look good to my manager." and "I must be the one to fix any bug I see – and I should fix it the 'right' way, even if I do not have time for it." [4].

In addition to technical skills and general misconceptions about industry, many students lack non-technical, "soft" skills such as oral and written communication, being able to work in a large group and understanding the ethics of their work [8, 27]. A study by Mardis et al. compared universities' syllabi to industry certifications, job postings and internship postings and found that many of the syllabi did not list any non-technical skills.

The motivation for remedying the gap is grounded in satisfying student career goals but also in our theoretical understanding of how students learn. Under Situated Learning, students learn better while working with others and they desire to be part of a community of professional practice [18]. Also relevant is Cognitive Apprenticeship where experts model behaviors that are meaningful to learners in a real-world context [7]. Learners then attempt to imitate those skills with coaching by the experts. Under Cognitive Apprenticeships, students want to learn from experts in their future discipline in a real-life context and this approach may enable students to achieve a high degree of learning [35].

2.2 Relationship Between Faculty and Undergraduate Beliefs

Some of the academia-industry gap—real or perceived—could be due to divergent views on CS education. Indeed, faculty and undergraduate attitudes and beliefs about computer science and computer science education do not always align. A study by Lewis et al. surveyed 13 faculty and 160 undergraduates on their beliefs and attitudes about a variety of CS-related topics, and then compared how student and faculty views differed. The survey questions were clustered into two topic areas: "Computer Science as Accomplishment" and "Computer Science as an Intellectual Discipline" [20, 21]. The survey results showed that while student and faculty views aligned in some cases, there were many areas of misalignment, including for students in their final year at the university. Several statements that were rejected by faculty and more accepted by students included "A significant problem in learning computer science is being able to memorize all the information I need to know", "Doing things the 'right' way is not as important as just pushing through to a solution", and "In the real world, computer scientists spend a lot of time working alone." These findings show that faculty and undergraduate students may not agree on important aspects of being a computer scientist, even when students are close to completing their degree. Such misconceptions could lead to related misconceptions as to what a computer science education should contain. On the other hand, such misconceptions could be influenced by how CS courses are conducted (e.g., testing for memorization and having students work alone). As will be shown below, if courses are run this way, it could be due to challenges in instruction rather than faculty beliefs about the goals of a CS education. Lewis et al.'s findings have been replicated and further explored, focusing on areas where faculty and students disagree [23].

It has been suggested that faculty designing curricula should understand student motives and program expectations in order to improve students' learning and performance [36]. A study aimed at identifying motives among geography majors found that students desire their curricula to focus more on career guidance and vocational training [11]. This is similarly true in computer science [8]. In some cases, students may not understand the relevance of many CS courses for their careers [24]. A recent study reported that half of CS students believed their CS program did not sufficiently prepare them for their professional experiences [16]. On the other hand, a study interviewing faculty found that many faculty believe CS programs do sufficiently prepare students for jobs in industry, stating that the majority of their graduates are able to find high paying jobs. However, faculty also cited topics like software maintenance as being difficult to teach and "soft" skills as under-taught [6].

Most recently, we conducted a study focused on faculty views on the goals of an undergraduate computer science education [34]. We conducted semi-structured interviews with 14 faculty members from three different universities, including a large research-focused university, a large undergraduate-focused university, and a small liberal arts university. Study participants came from a variety of technical backgrounds such as Theory, Machine Learning, and Bioinformatics. Coding the interviews via phenomenography [2], a wide-range of views were uncovered on the goals of a CS education, their institution's role in achieving those goals, and the barriers encountered in pursuing those goals. Separately, we also observed that our 14 faculty interviewees generally, although not universally, viewed industry preparation as an important part of an undergraduate degree in CS, yet many felt that they did not have the resources to make their courses successful in that regard. Our present study attempts to put these observations and related questions on a quantitative footing, asking similar questions in an online survey sent to faculty worldwide.

2.3 Barriers to Change

An element uncovered in our interview study was that faculty face barriers to narrowing the academia-industry gap [34]. There have been a number of studies investigating the challenges to changing faculty practices [14] and how best to address them (please see Henderson et al. for a summary of the literature [13]). Guidelines for fostering faculty change require identifying the barriers to adoption that faculty may face before attempting to create potential solutions. This allows solutions to account for those adoption challenges [17]. As such, this study seeks to identify which barriers are most commonly faced by faculty in order to inform future solutions.

3 Study Design

3.1 Research Questions

Based on the results of our prior qualitative study [34], we formulated the following research questions.

RQ1 What are the views of the broad community of CS faculty on the goals of an undergraduate CS education?

RQ2 What are the views of the broad community of CS faculty on the role of an undergraduate CS education in preparing students for industry?

RQ3 What are the views of the broad community of CS faculty on better preparing students for careers in industry?

RQ4 What barriers to providing more industry-relevant course content are the broad community of CS faculty encountering?

3.2 Methods

To gain insight on the above research questions, we selected relevant questions and findings from our prior qualitative study [34] and formulated them as survey questions. In order to assess whether the survey questions were being interpreted properly, we tested our survey on three graduate students in think-aloud sessions.

The final survey consisted of the following questions (a full version of the survey is posted online [1]):

(1) What course/area do you primarily teach or identify with?
(2) Do you teach at a public or private institution?
(3) Which of the following terms best defines your institution?
(4) In what country is your institution located?
(5) How would you rate the size of your institution in terms of estimated total number of undergraduate students?
(6) How would you rate the size of your CS program in terms of estimated number of undergraduate CS degrees awarded per year?
(7) What type of term system does your institution use?
(8) For each of the following items please list how important you believe this goal should be in your institution's undergraduate CS program.[1]
(9) Please indicate your agreement with the following statement: Preparing students for the academic challenges of a Masters or PhD program should be a principal goal of an undergraduate CS education.
(10) Please indicate your agreement with the following statement: Preparing students for careers in industry should be a principal goal of an undergraduate CS education.
(11) Out of a total of 100%, what do you believe should be the balance of responsibilities for preparing students for the academic challenges of their Masters or PhD programs?
(12) Out of a total of 100%, what do you believe should be the balance of responsibilities for preparing students for careers in industry?
(13) Please indicate your agreement with the following statement: It would be beneficial for the quality of my institution's CS program if we investigated how we could better prepare our students for careers in industry.
(14) Please list your agreement with the following statements. In order to better prepare students for their future careers than we are doing today at my institution, I believe we should...[1]
(15) Please indicate your agreement with the following statements. The strongest barriers to me providing more directly industry-relevant content in my primary course are...[1]
(16) What is your gender?
(17) What is your age group?
(18) Did one (or more) of your parents/caregivers receive a college or university diploma?
(19) Do you identify with a group that is underrepresented in computer science programs in your country? (e.g. ethnicity)

3.3 Dissemination of Survey

We employed a range of methods to maximize the reach of our survey. First, we posted our survey invitation to a few established CS community mailing lists, although we were unable to send to many mailing list as most do not allow the posting of surveys. Second, using automated methods, we scraped information from Google Scholar by searching for profiles with tags related to computer science. Finally, we similarly scraped the faculty listings of

[1] The full item list can be found online [1]

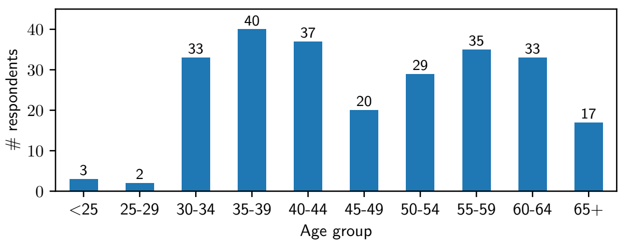

Figure 1: Q17: Age distribution of respondents.

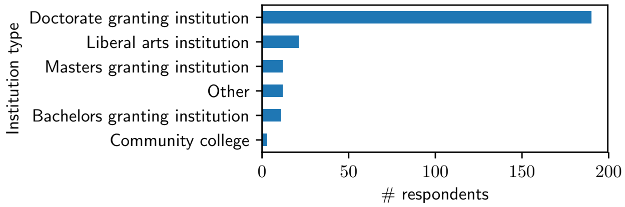

Figure 2: Q3: Institution type of respondents.

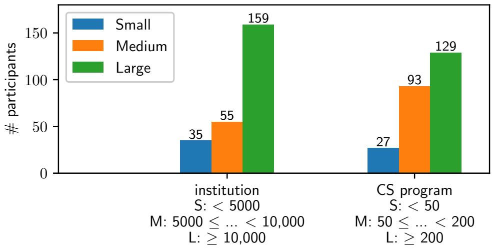

Figure 3: Q5-6: Institution and CS program sizes by number of undergraduates.

a wide selection of computer science department web sites. We obtained these department websites by searching on Google for the institution names as they appeared in the Times Higher Education Ranking [33] together with the words "computer science department". In all, we emailed approximately 7250 faculty at 140 institutions in 35 countries.

An example of the email template used for participant recruitment can be found online [1].

4 Results

We received 325 responses in total, 249 of which are complete responses (77% completion rate). Unless stated otherwise, the results we report are based on the 249 complete surveys.

4.1 Demographics

Of the 249 respondents, 200 were men, 37 were women, 2 were non-binary, and 10 chose prefer not to answer. The age distribution of the respondents is shown in Figure 1. A total of 66 respondents were first-generation students and 40 respondents identified with an underrepresented group in CS, including gender, sexual orientation, ethnicity, disability, socioeconomic status and age.

As illustrated by Figure 2, the great majority of our respondents are from doctoral granting institutions. We also found that 70% of respondents are from public institutions and 30% work at private institutions. The great majority of respondents work with a semester system (74%), followed by the quarter system (13%), and the trimester system (10%). The remaining 3% use other term systems. Figure 3 shows that most participants are from large schools with

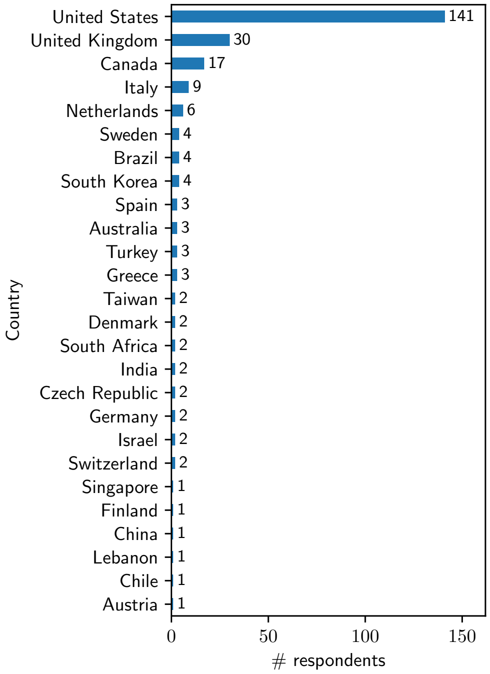

Figure 4: Q4: Country distribution of respondents.

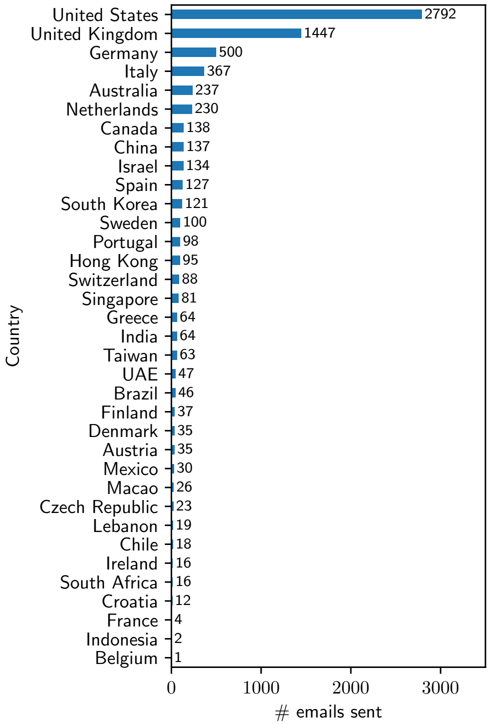

Figure 5: The number of emails sent per country.

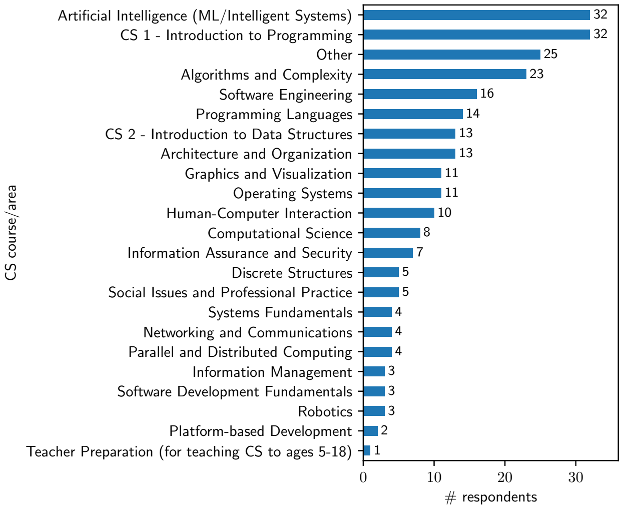

Figure 6: Q1: CS course/area with which faculty respondents primarily identify themselves. Courses and areas taken from the ACM curricula guidelines [15].

medium to large CS programs. Responses were collected from 26 countries in all.

As seen in Figure 4, the majority of our respondents are from the United States (57%). Figure 5 shows that while did send out a larger number of emails to the United States than to any other country, it was not the majority of our emails (39%). However, the response rate for emails sent to the United States at 5% was higher than the average response rate of 3%.

Around 13% of our participants identified CS1 as their primary course. The full distribution of faculty respondents' primary courses/ areas can be found in Figure 6.

4.2 Faculty Views on the Goals of CS

With the exception of just two people who voted "Not at all important" and one person who voted "Slightly important", faculty overwhelmingly support "For students to become competent problem solvers" as a goal of a CS program. Other strongly supported goals include "For students to learn how to program", "For students to know how to learn a new technology or algorithm quickly", "For students to attain 'hard' skills" and "For students to attain a breadth of CS knowledge". The vast majority of the listed goals received only a very small number of "Not at all important" votes, with "For students to figure out what their next step in life should be" as the item receiving the most dissidence. This high level of agreement is partially to be expected because the stated goals were taken from those expressed by faculty in our previous qualitative study [34].

4.3 Views on Industry-Specific Program Goals

Figure 7 shows that, while there is a minority who disagree, most faculty support industry specific goals such as developing strong software engineering skills and preparing students for the job market. In an interesting contrast, faculty seem to be split on the importance of keeping the program up to date with industry practice. This could be due to faculty perceptions that software industry trends are not always driven by fundamental changes to practice, but rather by factors like cost, tool compatibility, or levels of scalability not relevant to the course context.

Figure 8 shows that faculty seem to have a stronger agreement that industry preparation is a principal goal of a CS program. The medians were "Agree" for industry preparation and "Neutral" for academic preparation. We ran a one-sided Mann-Whitney U test to determine whether or not the agreement was significantly greater for industry preparation. We found significance for $\alpha = 0.05$ with a Mann-Whitney U statistic of 38792.0 and a p-value of 1.935e-07.

More importantly, Figure 9 shows that for the preparation of students for careers in both industry and academia, faculty view their own institutions as the primary responsible party. This is encouraging as finding other results here would likely imply that faculty are not willing to close the academia-industry gap. Faculty do, however, place a significant portion (around 25%) of the responsibility on both the students themselves and the party hiring the students after graduation. For industry preparation, faculty expect the companies hiring the students to take on a larger part of the remaining responsibility whereas for preparation for academia, faculty expect the students to take on a larger part of the remaining responsibility.

We also explored the role that age has on faculty perceptions of the value of industry preparation. Our hypothesis was that older faculty may have entered CS when it was part of mathematics departments and hence more theory-heavy, perhaps making older faculty less open to teaching content relevant for industry preparation. As such, we grouped faculty by age, binning those under 50 years old as younger. Comparing medians led to the interesting finding that, contrary to our expectation, older faculty had a higher median rating for industry preparation as a principal goal ("Agree", vs. "Neutral" for early-career faculty). And indeed when running a one-sided Mann-Whitney U test to establish significance, we find that older faculty reported a significantly greater agreement with industry preparation as a principal goal for $\alpha = 0.05$ with a statistic of 6406.5 and a p-value of 0.008.

For preparation for academia as a principal goal, we find no significant differences between younger and older faculty. Both groups report a median of "Neutral". A two-sided (to identify any difference in either direction) Mann-Whitney U test led to a statistic of 7475.5 and a p-value of 0.343, indicating no significant differences for $\alpha = 0.05$.

4.4 What Should Be Improved?

The majority of faculty believe their curriculum should be improved with respect to preparing students for industry careers, as can be seen in Figure 10. This finding contradicts popular claims that "ivory tower" faculty attitudes would be responsible for perpetuating the academia-industry gap. The same graph also shows that most of the needs for improvements that surfaced from our prior interviews [34] can count on broad support among faculty. The strongest support can be found for integrating software engineering best practices, such as software testing, more explicitly in courses with a programming component. Faculty appear to be split on the topics of reducing grade inflation and reducing class sizes. Overall, we see broad support for many ideas for improvement, but for each of them there is also a number of faculty who oppose this idea. In Figure 11, we also see broad support from faculty for investigating

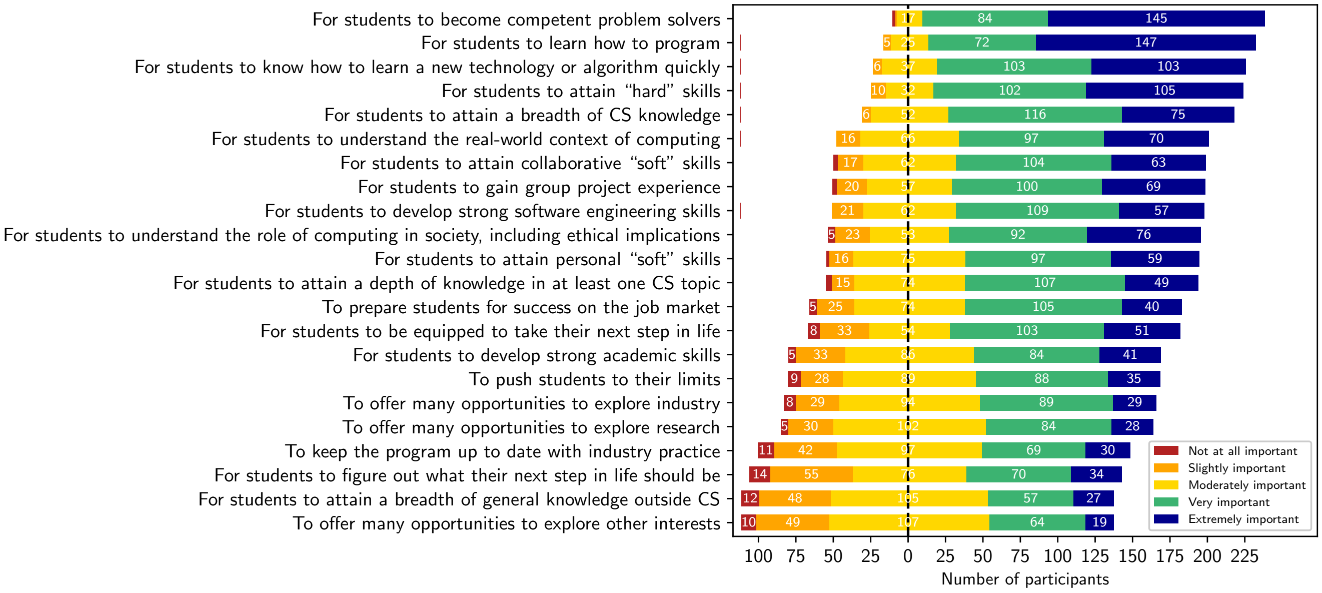

Figure 7: Q8: Relative support for goals of a CS program, "For each of the following items please list how important you believe this goal should be in your institution's undergraduate CS program."

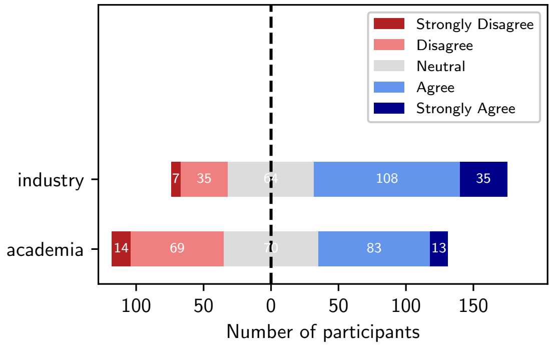

Figure 8: Q9-10: Relative support for industry preparation and academic preparation as a principal goal of a CS program. "Preparing students for [the academic challenges of a Masters or PhD program/careers in industry] should be a principal goal of an undergraduate CS education"

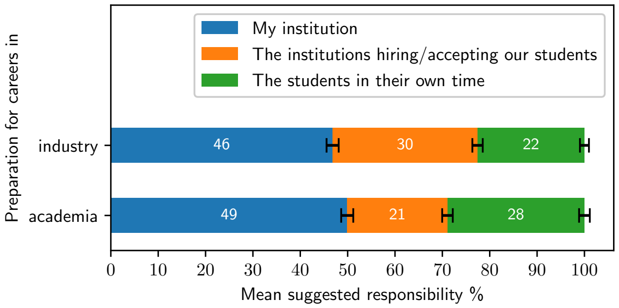

Figure 9: Q11-12: Balance of responsibilities in preparing students for careers in academia/industry according to faculty. "Out of a total of 100%, what do you believe should be the balance of responsibilities for preparing students for [the academic challenges of their Masters or PhD programs/careers in industry]?"

how their institution could better prepare students for industry careers.

4.5 Barriers to Improvement

Figure 12 shows that faculty appear to be quite split in terms of which barriers limit their ability to provide more industry-relevant content in their courses. Although there are perceived barriers, we see a continued trend in desiring to better prepare students for industry in the fact that the majority of faculty do not oppose including industry-relevant content in their courses. However, there is a sizeable group of faculty who agree or strongly agree that such content does not belong in their course. This difference in the number of faculty supporting better industry preparation from the number who support doing so in their own courses may be a factor in perpetuating the academic-industry gap. As such, we wanted to determine if faculty from certain courses or areas of CS are more likely than others to find that industry content does or does not belong in their course. Figure 13 explores participant agreement with this statement based on the course they most commonly teach. From this figure, it appears that instructors who teach CS1, Software Engineering or Human-Computer Interaction are the most supportive of teaching industry-relevant content in their course. In

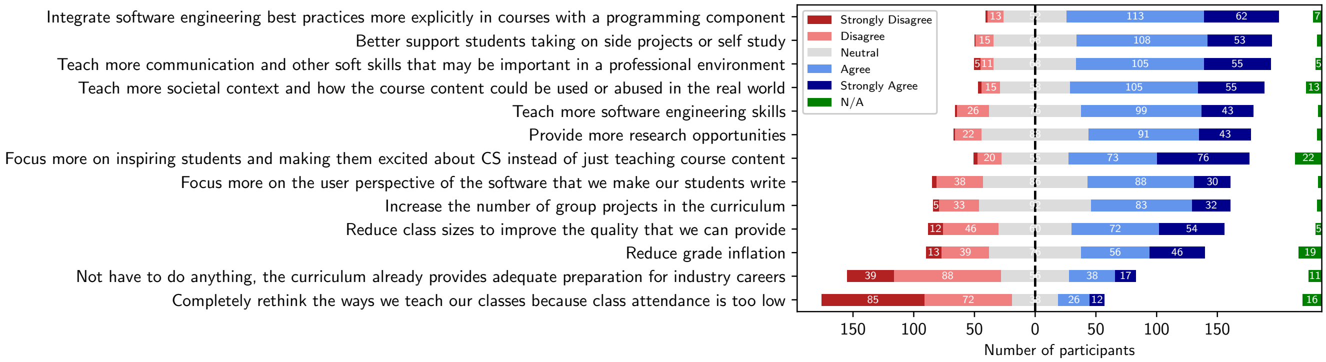

Figure 10: Q14: Faculty views on better preparing students for industry. "Please list your agreement with the following statements. In order to better prepare students for their future careers than we are doing today at my institution, I believe we should..."

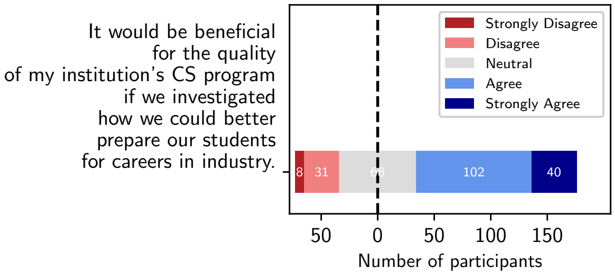

Figure 11: Q13: Relative support for investigating how to better prepare students for industry. "It would be beneficial for the quality of my institution's CS program if we investigated how we could better prepare our students for careers in industry."

contrast, instructors who teach Algorithms and Complexity represent the largest group opposing industry-relevant content in their courses. Somewhat surprisingly though, we also find a sizeable portion of Programming Languages faculty believe such content does not belong in their course.

Since faculty appear to be split on whether large class size is a barrier and on the need to reduce class sizes to improve the quality of education, we decided to compare the results on these two survey items for faculty from small and large institutions.

We found that the median for "The class size is too large" as a barrier to improvement to be "Disagree" for faculty from small CS departments, whereas faculty from large CS departments responded with a median of "Neutral". For "Reduce class sizes to improve the quality that we can provide" as an improvement, the median was "Neutral" for faculty from small CS departments and "Agree" for faculty from large CS departments.

We followed up by running a one-sided Mann-Whitney U test to determine whether the responses for faculty from larger CS departments were statistically greater than responses from faculty from smaller CS departments for these two survey items. We found "The class size is too large" had a Mann-Whitney U statistic of 2369.0 and a p-value of 0.001, indicating that the response for faculty from large CS departments was significantly higher for $\alpha = 0.05$. For "Reduce class sizes to improve the quality that we can provide", we found a Mann-Whitney U statistic of 2349.0 and a p-value of 0.002, also indicating a significantly greater agreement for faculty from larger CS departments for $\alpha = 0.05$.

Another interesting point to note is the large numbers of "N/A" responses for the questions regarding instructional staff in Figure 12. Specifically, for the item "My instructional staff (e.g. teaching assistants) is not sufficiently aware of industry best practices", 41% of the faculty from small CS departments responded with "N/A". We suspect the reason for this is that due to smaller class sizes they may not have instructional staff available to them.

Figure 14 shows that there is not a two group split between one group of faculty who experience all the barriers and another group of faculty who experience no barriers. While there is a sizable group of faculty who experience no barriers at all, the vast majority of faculty experience at least some barriers to providing more directly industry-relevant content in their courses.

5 Discussion

Concerning RQ1, we find that faculty demonstrate strong support for the following goals of an undergraduate CS education:

(1) For students to become competent problem solvers
(2) For students to learn how to program
(3) For students to know how to learn a new technology or algorithm quickly
(4) For students to attain "hard" skills
(5) For students to attain a breadth of CS knowledge

Whereas faculty are in disagreement about the importance of the following goals:

(1) To keep the program up to date with industry practice
(2) For students to figure out what their next step in life should be
(3) For students to attain a breadth of general knowledge outside CS
(4) To offer many opportunities to explore other interests

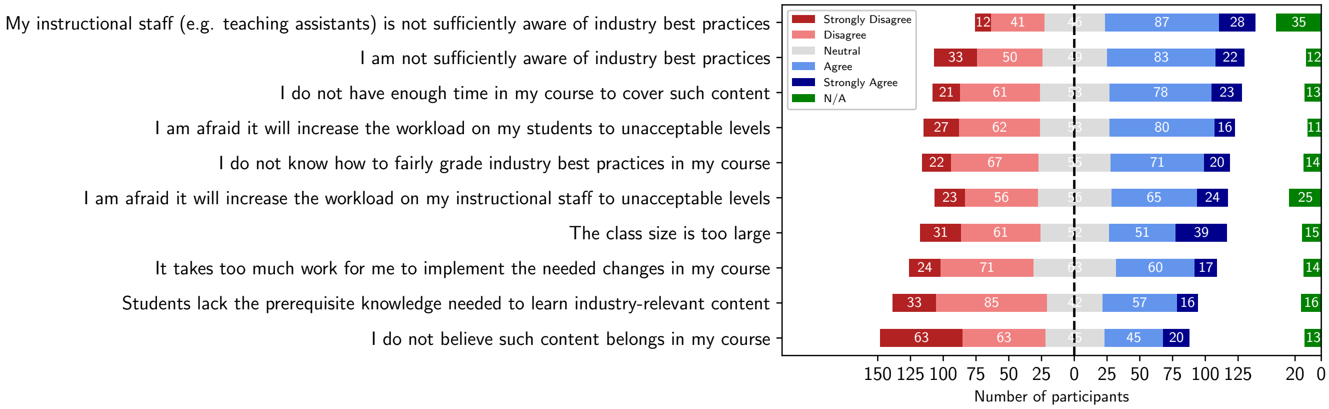

Figure 12: Q15: Barriers experienced by faculty. "Please indicate your agreement with the following statements. The strongest barriers to me providing more directly industry-relevant content in my primary course are"

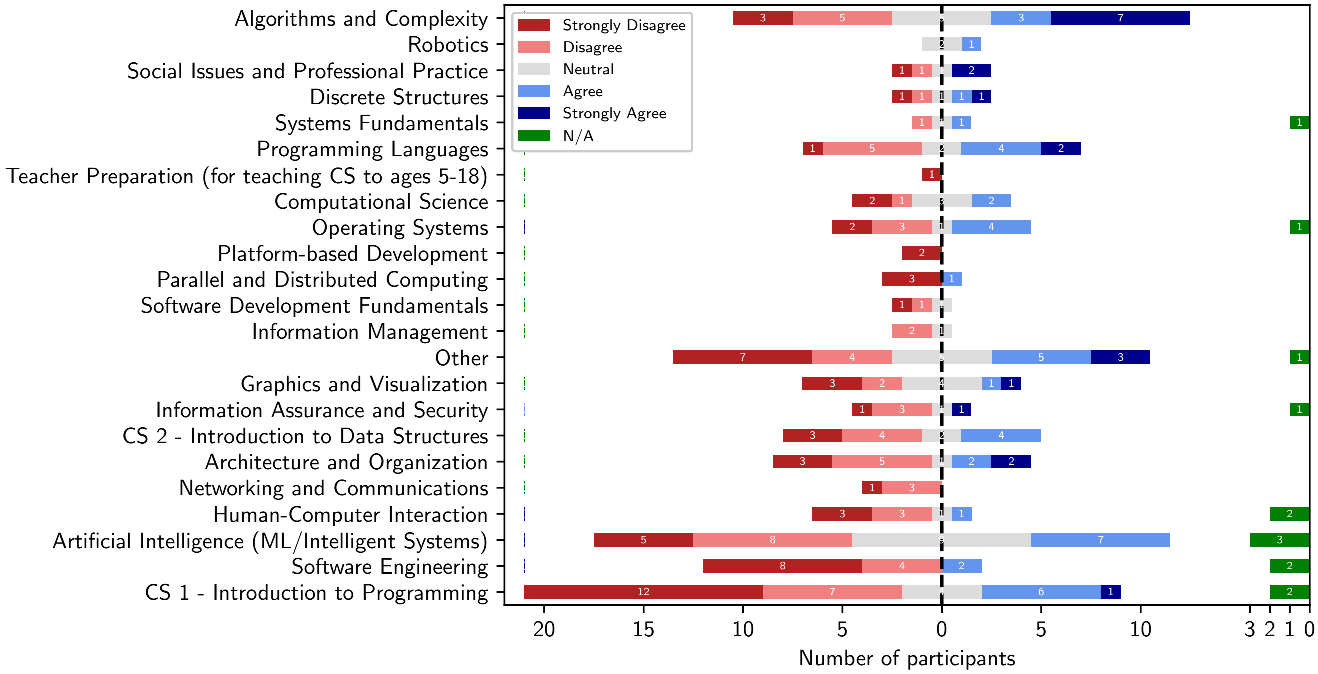

Figure 13: Q15: Participant level of agreement with "I do *not* believe such content belongs in my course" as a barrier to providing directly industry-relevant content in their course, plotted against the main CS course/area they identify with. Courses and areas taken from the ACM curricula guidelines [15].

Regarding RQ2, we find that 57% of faculty agree that industry preparation is a principal goal of an undergraduate CS education whereas only 17% of faculty disagree. The faculty opinion is more split on whether or not preparing students for academia is a principal goal (39% agree, 33% disagree). Moreover, faculty view their own institution as the primary responsible party for preparing students for both industry and academia.

For RQ3, we find that a majority of faculty believes it would be beneficial for the quality of their institution's CS program if they would investigate how they could better prepare students for careers in industry (57% agree, 16% disagree). We also find that many faculty expressed support for improving their courses by integrating materials and experiences related to industry preparation into their courses. Examples of such materials are software engineering best practices and soft skills. However support for such course changes varied widely by course area.

With respect to RQ4, we find that faculty were split regarding the number of barriers encountered as well as which barriers they

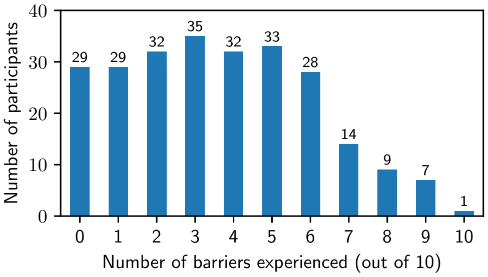

Figure 14: Q15: Number of barriers experienced per participant (counting number of "Agree" and "Strongly Agree" ratings)

encountered. Unfortunately this finding implies that the academia-industry gap is a multi-faceted problem that most likely cannot be solved by addressing a single barrier.

The remainder of this section explores these research questions in greater depth.

5.1 Dissenting Views

Although a majority of our respondents agree that preparing students for industry should be a primary goal of undergraduate CS education and that the quality of their institution's CS program would be improved by better preparing students for industry, there is a substantial group of faculty who disagree with these statements. These faculty may believe, as the quote attributed to a famous CS professor in Stroustrup [32]: "We don't teach programming; we teach computer science", that software engineering skills important for industry are different than the skills of a computer scientist. These beliefs are apparent in the distinction between software engineering [3] and computer science [28] degree programs. What matters most, perhaps, is that these disagreements about the goals of CS education exist. If institutions are working to improve student preparation for industry, these dissenting voices could slow progress and represent a new barrier not currently addressed in our survey.

5.2 Entrenched Beliefs?

One might wonder if those faculty who have been in the present system longer would be more resistant to change, in particular change towards aligning CS curricula with better industry preparation. This could potentially explain the persistence of the gap, i.e., younger faculty are eager to make changes, while entrenched interests resist that change, causing the younger faculty to lose momentum.

However, as discussed in Section 4.3 we found that older faculty were be more inclined to support industry preparation as a primary goal. Thus entrenched beliefs, if present, would appear to support the goal of industry preparation. Interestingly, Florian et al. found that employee age is negatively correlated with resistance to change [10]. This result would support the idea that older faculty, having greater support for industry preparation, would be more open to curriculum changes that improve industry preparation.

Why older faculty report higher agreement with industry preparation as an important programmatic goal remains unclear. However, we wonder if it might be that faculty learn more about the merits of an industry background throughout their interactions over the years with graduated students or with people returning to academia from industry. They may also be more likely to have older children who are making career choices; having a concern for their success could cause a broader pragmatic turn.

5.3 First-Generation Respondents

In contrast with previous work which finds first-generation students have a stronger job-focused mentality [19], we did not find any statistically significant difference for faculty who were first-generation students and their agreement on the importance of industry preparation as an important goal. We find that for the item "To prepare students for success on the job market" as a goal of CS programs, both faculty who were first-generation students and faculty who were not first-generation students report a median of "Very important". A two-sided Mann-Whitney U test reported a statistic of 5630.0 and a p-value of 0.426, indicating no significant difference for $\alpha = 0.05$.

Why there appears to be a difference in job-focused mentality between first-generation students and faculty who used to be first-generation students remains unclear. We wonder if perhaps faculty having tenured jobs could play a role here or if the subset of students who pursue faculty positions is different from the general population of first-generation students. Future work could investigate these questions.

5.4 CS Subdisciplines

We find some evidence that certain areas of CS may be more biased than others against industry preparation. As discussed in Section 4.5, we find that faculty identifying with the areas of Algorithms and Complexity as well as Programming Languages are more likely to believe that industry-relevant content does not belong in their course. However, we should keep in mind that a possible explanation for this is that it might be the case that academia is ahead of industry in these specific fields. For example, one participant who identifies with the Programming Languages area reached out to us over email after completing the survey. They stated that it would be counter-productive to cover directly industry-relevant content in their course because it would mean not covering the state of the art in their field. According to this participant, it was not necessarily the case that such content does not belong in their course, but more that they believed it would be counter-productive to inspiring students and preparing them for life-long learning. It is unclear how many of our respondents from these areas may share this perspective.

5.5 Barriers Encountered

As demonstrated in Section 4.5, faculty face barriers towards providing more directly industry-related content in their courses. Identifying common barriers is important as they inherently impact the likelihood that faculty will adopt possible solutions [13, 17]. We already see some evidence of this because the academia-industry gap persists despite the many papers that have been published on how to effectively teach industry-relevant content. For instance, there is a plethora of studies on how to teach software testing as well as on how to automatically grade it; some examples are [9, 29, 31]. Unfortunately, these practices are not yet commonplace at many institutions (among which our own) which suggests these barriers may be inhibiting adoption. Future work is needed to investigate why such studies have not gained traction and whether or not that may be related to the barriers discussed in Section 4.5.

Furthermore, we find in Section 4.5 that not all faculty experience the same barriers. Figure 14 demonstrates that we find about half of the faculty experience relatively few barriers while about a quarter experience substantial barriers. For example, we unsurprisingly find that faculty from schools with larger CS programs experience class size as a barrier to improvement. Similarly, for smaller CS programs, their faculty report experiencing fewer barriers pertaining their instructional staff as they may not have any. For most barriers, however, it remains unclear why certain faculty are experiencing these barriers while others are not. Future work is needed to uncover what separates these groups of faculty for each barrier such that the barrier can be addressed.

5.6 Research and Policy Implications

The results of this study have implications for researchers as well as administrators. For instance, we find substantial support for industry preparation among CS faculty, yet the academia-industry gap in CS remains. Furthermore, faculty cite experiencing substantial and diverse barriers that obstruct the way towards providing more industry-relevant content in their courses. We do not find one or two clear barriers that almost all faculty experience. Instead we find that each of the barriers in Figure 12 is experienced by a substantial number of faculty. These findings indicate that the academia-industry gap is a multi-faceted issue and that more work is needed in understanding and overcoming these barriers.

6 Limits and Risks to Validity

As a closed-ended survey study, the range of responses to questions was limited to the responses previously elicited in our open-ended interviews [34]. Although there may have been other responses that respondents may have wished to choose, we do note that our questions achieved a sufficiently wide range of responses to give us confidence that the responses to our survey represented real alternatives and choices for the respondents.

Our survey results are subject to potential selection bias. Participation in our survey was voluntary, without remuneration, so there could be a bias in our results due to the type of person who answers such surveys (e.g., respondents may be more community-minded than the average faculty member). We attempted to counter other selection biases by using multiple methods to target respondents and building department mailing lists from a wide variety of institutions.

Related to the above, although we solicited responses from thousands of faculty, we achieved a response rate of only 3%, a relatively low proportion, although typical for survey studies. For example, Sheehan reports the following factors are strongly correlated to email survey response rates: Survey Length, Respondent Pre-Notification, Follow-Up Contact/Reminders, and Relevance of the Issue to the People Approached [30]. Our survey does not do well on these factors, except for the relevance of the issue to the people we approached. We could of course have sent pre-notifications or reminders, however we did not want to bother people with more than one unsolicited email about our survey. Our completion rate of 77% gives us confidence that the survey design itself was not deterring people from completing it.

We used statistical methods where appropriate in order to test for confidence, and were clear on when we did and did not achieve sufficient confidence.

Furthermore, although we solicited responses from 35 countries, a large portion of our participants are from universities from the US and other English speaking countries. This may limit the generalizability of our results.

As a survey study, our results reflect what respondents consciously chose as answers in the survey, not necessarily what they actually believe or enact in their daily lives. While an observational study (e.g., of department faculty meetings or the conduct of courses) might reveal actual beliefs or tendencies, it would be extraordinarily difficult to achieve the quantitative results required to lend insight on the research questions of this study. On the other hand, because survey responses were anonymous, there was no incentive for our respondents to intentionally mislead.

7 Conclusion

Although the academia-industry gap has been long-studied, little was known about the perspectives of the broad community of CS faculty. In our survey study of 249 faculty, we find that the community is strongly supportive of industry preparation as a goal of their undergraduate CS programs. Moreover, they generally view their own institution as the primary party responsible for industry preparation. Faculty are also strongly supportive of teaching more industry-related content, for example by integrating more software engineering practices into existing courses and by teaching more soft skills that may be important in a professional context. That said, this support varies by the faculty's primary course area, with instructors of software-related courses offering more support and instructors of more mathematical courses offering less support.

The enthusiasm for the goal of industry preparation is tempered by an acknowledgment that there are many barriers to success. Although faculty varied widely in the number of barriers they cited as preventing their inclusion of more industry-related content, over half encountered more than three barriers. There were no discernible patterns in the number of barriers according to institution or program characteristics.

With such strong faculty support for industry preparation, the common barriers uncovered in this work may offer a partial answer to the question of why the gap persists. A clear research priority should then be learning more about the characteristics of these barriers and how we might begin to remedy them.

8 Acknowledgements

The authors thank the participating faculty for making time in their busy schedules to fill out our survey as well as the reviewers and the faculty who responded to our email for their thoughtful suggestions. This work was supported in part by NSF award DUE-1712508. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650112. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Faculty Survey. https://paperdata.page.link/faculty-survey

[2] Gerlese S. Åkerlind. 2005. Variation and commonality in phenomenographic research methods. *Higher Education Research & Development* 24, 4 (2005), 321–334. https://doi.org/10.1080/07294360500284672

[3] 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* 11 (2015), 106–109.

[4] Andrew Begel and Beth Simon. 2008. Struggles of New College Graduates in Their First Software Development Job. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '08)*. ACM, New York, NY, USA, 226–230. https://doi.org/10.1145/1352135.1352218

[5] Gail Carmichael, Christine Jordan, Andrea Ross, and Alison Evans Adnani. 2018. Curriculum-Aligned Work-Integrated Learning: A New Kind of Industry-Academic Degree Partnership. In *Proceedings of the 49th Special Interest Group on Computer Science Education Technical Symposium*. 586–591.

[6] Secil Caskurlu, Iryna Ashby, and Marisa Exter. 2017. The Alignment Between Formal Education and Software Design Professionals' Needs in Industry: Faculty Perception. In *2017 ASEE Annual Conference & Exposition*. ASEE Conferences. https://peer.asee.org/28941.

[7] Allan Collins, John Seely Brown, and Ann Holum. 1991. Cognitive apprenticeship: Making thinking visible. *American educator* 15, 3 (1991), 6–11.

[8] Michelle Craig, Phill Conrad, Dylan Lynch, Natasha Lee, and Laura Anthony. 2018. Listening to Early Career Software Developers. *J. Comput. Sci. Coll.* 33, 4 (April 2018), 138–149. http://dl.acm.org/citation.cfm?id=3199572.3199591

[9] Stephen H. Edwards. 2003. Rethinking Computer Science Education from a Test-first Perspective *(OOPSLA '03)*.

[10] Kunze Florian, Boehm Stephan, and Bruch Heike. 2013. Age, resistance to change, and job performance. *Journal of Managerial Psychology* 28, 7/8 (01 Jan 2013), 741–760. https://doi.org/10.1108/JMP-06-2013-0194

[11] Sharon Gedye, Elizabeth Fender, and Brian Chalkley. 2004. Students' Undergraduate Expectations and Post-graduation Experiences of the Value of a Degree. *Journal of Geography in Higher Education* 28, 3 (2004), 381–396. https://doi.org/10.1080/0309826042000286956

[12] Hisham Haddad. 2002. Post-graduate Assessment of CS Students: Experience and Position Paper. *J. Comput. Sci. Coll.* 18 (2002).

[13] Charles Henderson, Andrea Beach, and Noah Finkelstein. 2011. Facilitating change in undergraduate STEM instructional practices: An analytic review of the literature. *Journal of research in science teaching* 48, 8 (2011), 952–984.

[14] Charles Henderson and Melissa H Dancy. 2007. Barriers to the use of research-based instructional strategies: The influence of both individual and situational characteristics. *Physical Review Special Topics-Physics Education Research* 3, 2 (2007), 020102–1–020102–14.

[15] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA. https://doi.org/10.1145/2534860

[16] Amanpreet Kapoor and Christina Gardner-McCune. 2019. Understanding CS Undergraduate Students' Professional Development Through the Lens of Internship Experiences. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, 852–858. https://doi.org/10.1145/3287324.3287408

[17] Raina Khatri, Charles Henderson, Renée Cole, Jeffrey E Froyd, Debra Friedrichsen, and Courtney Stanford. 2016. Designing for sustained adoption: A model of developing educational innovations for successful propagation. *Physical Review Physics Education Research* 12, 1 (2016), 010112.

[18] Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.

[19] Wolfgang Lehmann. 2009. University as vocational education: working-class students' expectations for university. *British Journal of Sociology of Education* 30, 2 (2009), 137–149. https://doi.org/10.1080/01425690802700164

[20] Clayton Lewis. 2007. Attitudes and Beliefs About Computer Science Among Students and Faculty. *SIGCSE Bull.* 39, 2 (June 2007), 37–41. https://doi.org/10.1145/1272848.1272880

[21] Clayton Lewis, Michele H. Jackson, and William M. Waite. 2010. Student and Faculty Attitudes and Beliefs About Computer Science. *Commun. ACM* 53, 5 (May 2010), 78–85. https://doi.org/10.1145/1735223.1735244

[22] Marcia A. Mardis, Jinxuan Ma, Faye R. Jones, Chandrahasa R. Ambavarapu, Heather M. Kelleher, Laura I. Spears, and Charles R. McClure. 2018. Assessing alignment between information technology educational opportunities, professional requirements, and industry demands. *Education and Information Technologies* 23, 4 (01 Jul 2018), 1547–1584. https://doi.org/10.1007/s10639-017-9678-y

[23] Jacob Perrenet. 2009. Differences in Beliefs and Attitudes About Computer Science Among Students and Faculty of the Bachelor Program. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*. ACM, 129–133. https://doi.org/10.1145/1562877.1562920

[24] Anne-Kathrin Peters, Anders Berglund, Anna Eckerdal, and Arnold Pears. 2015. Second Year Computer Science and IT Students' Experience of Participation in the Discipline. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research (Koli Calling '15)*. ACM, New York, NY, USA, 68–76. https://doi.org/10.1145/2828959.2828962

[25] R. Pham, S. Kiesling, L. Singer, and K. Schneider. 2016. Onboarding inexperienced developers: struggles and perceptions regarding automated testing. *Software Quality Journal* 25, 4 (2016), 1239–1268. https://doi.org/10.1007/s11219-016-9333-7

[26] 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)*. ACM, New York, NY, USA, 525–530. https://doi.org/10.1145/2445196.2445351

[27] Alex Radermacher, Gursimran Walia, and Dean Knudson. 2014. Investigating the Skill Gap Between Graduating Students and Industry Expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering (ICSE Companion 2014)*. ACM, New York, NY, USA, 291–300. https://doi.org/10.1145/2591062.2591159

[28] Mehran Sahami and Steve Roach. 2014. Computer science curricula 2013 released. *Commun. ACM* 57, 6 (2014), 5–5.

[29] Zalia Shams. 2013. Automated Assessment of Students' Testing Skills for Improving Correctness of Their Code *(SPLASH '13)*.

[30] Kim Bartel Sheehan. 2001. E-mail Survey Response Rates: a Review. *Journal of Computer-Mediated Communication* 6, 2 (01 2001). https://doi.org/10.1111/j.1083-6101.2001.tb00117.x JCMC621.

[31] Sulayman K. Sowe, Ioannis Stamelos, and Ignatios Deligiannis. 2006. A Framework for Teaching Software Testing using F/OSS Methodology. In *Open Source Systems*, Ernesto Damiani, Brian Fitzgerald, Walt Scacchi, Marco Scotto, and Giancarlo Succi (Eds.). Springer US, Boston, MA, 261–266.

[32] Bjarne Stroustrup. 2010. What Should We Teach New Software Developers? Why? Vol. 53 № 1. *Commun. ACM* (2010), 40–42.

[33] Times Higher Education Ranking. Times Higher Education Ranking. https://www.timeshighereducation.com/world-university-rankings/2020/subject-ranking/computer-science

[34] Sander Valstar, Sophia Krause-Levy, Alexandra L. 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 Special Interest Group on Computer Science Education Technical Symposium*. 577–583.

[35] Lev Vygotsky. 1978. Interaction between learning and development. *Readings on the development of children* 23, 3 (1978), 34–41.

[36] Anna Wilson, Susan Howitt, Pam Roberts, Gerlese Åkerlind, and Kate Wilson. 2013. Connecting expectations and experiences of students in a research-immersive degree. *Studies in Higher Education* 38, 10 (2013), 1562–1576. https://doi.org/10.1080/03075079.2011.633163