Understanding the Software Development Industry's Perspective on Coding Boot Camps versus Traditional 4-year Colleges

Quinn Burke
College of Charleston
burkeqq@cofc.edu

Cinamon Bailey
College of Charleston
kerrcs@g.cofc.edu

Louise Ann Lyon
ETR Associates
louann.lyon@etr.org

Emily Green
ETR Associates
emily.green@etr.org

ABSTRACT

This study reports on the perceived capacity of coding boot camps versus computer science (CS) undergraduate programs to instill a range of software development skills from an industry perspective. We present findings from a series of national focus groups and individual interviews with representatives from the software development industry, who spoke about their hiring procedures and preferences as well as how they perceived coding boot camp applicants in comparison to graduates from four-year CS degree programs. We also present findings on how the boot camp and university participants viewed their role in developing necessary skills for employability. Results indicate that hiring managers filling positions, generally have a favorable perspective of coding boot camp hires in relation to their demonstration of "soft" skills, such as teamwork, passion, and persistence; With regards to four-year university hires, several industry representatives indicated that a four-year degree is mandatory for hire, while also listing a solid understanding of CS principles and substantial exposure to mathematics. The Discussion section focuses on the future potential of coding boot camps as an alternative training ground for the software development industry.

KEYWORDS

Coding boot camps; industry needs; experiential learning

ACM Reference Format

Q. Burke, C. Bailey, L. A. Lyon, and E. Green. 2018. Understanding the Software Development Industry's Perspective on Coding Boot Camps versus Traditional 4-year Colleges. In *SIGCSE '18: 49th ACM Technical Symposium on Computer Science Education, Feb. 21–24, 2018, Baltimore, MD, USA.* ACM, NY, NY, USA, 6 pages. https://doi.org/10.1145/3159450.3159485

SIGCSE'18, February 21–24, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5103-4/18/02…$15.00
https://doi.org/10.1145/3159450.3159485

1. INTRODUCTION

One of the most prominent challenges facing U.S. higher education in the STEM fields is meeting the ever-growing computing needs of the U.S. workforce. The U.S. Bureau of Labor Statistics [1] projects the United States will have over one-million unfulfilled software engineering jobs by 2020, with the non-profit Code.org [2] listing the current openings in the U.S. at over 600,000. While post-secondary computer science enrollments have steadily risen since 2006 [3], the country is still drastically under-producing the computing literate workforce that an ever-growing number of industries demand.

Into this gap, a new educational player has emerged. Multi-week coding schools, such as CoderFoundry (http://www.coderfoundry.com) and Coding Dojo (http://www.codingdojo.com), offer students intensive introductions to coding, with the goal that their graduates can land entry-level software development positions after a relatively brief, yet intensive, introductory course (10-12 weeks on average according to Course Report [4]). The price tag for such education represents a steep upfront cost at $12,147 on average [4]. The average cost of a 4-year degree currently at $98,440 [5].

Most information we have about these camps comes from popular media articles detailing "rags to riches" success stories [6, 7] or cautionary tales about hopeful students, who are cheated by bad operators in the field [8]. Our exploratory research intends to move beyond the rather casual sketches of coding boot camps offered by popular media and further delve into the role of coding boot camps in the growing post-secondary STEM education market by examining the software development industry's perspective of these programs and how they compare and fit in with traditional 4-year degree programs in CS. This paper examines what particular skill sets industry representatives expect of their entry level software developers, and to what degree coding boot camps and 4-year degree programs are meeting such needs. After a review of the relevant literature around the requisite mix of "hard" and "soft" skills characteristic of effective software developers, this paper reports on the findings of a series of national focus groups and interviews with software development representatives, boot camp representatives, and university representatives. We also

discuss potential limitations of the current research as well as detail our next steps, particularly with the addition of boot camp and college student perspectives to offer a counter perspective to industry's professed outlook.

2. BACKGROUND

In order to pinpoint skills that are necessary for employability and career success, it is critical to have clear definitions of these skills. In their paper "What Makes a Great Software Engineer?" Li, Ko, and Zhu [9] arrived at a total of 53 distinct attributes that mark exceptional software engineers and noted that across these attributes, effective software engineering is clearly not just about content productivity but likewise about the manner with which such content is produced (i.e., the "how" is stressed much more often by interviewees than the "what").

Within industry, this distinction between "how" and "what" often translates into the distinction between "soft" and "hard" skill sets. "Soft" skills refer to those skills that relate to a person's personality and ability to work with others (communication, critical thinking, empathy, conflict resolution, flexibility, adaptability, and creativity among others). Yet despite the poplar moniker of "soft", such skills are increasingly being identified as the crucial attributes for success in the current marketplace, with prominent institutions [10], Meanwhile, "hard" skills refer to specific knowledge and cognitive capabilities required for a job (computer programming, mathematics, data analysis, etc.). With regards to "hard" skills, an individual may possess procedural knowledge (how something works in a certain way) and/or conceptual knowledge (why something works the way it does) [11].

Hatano and Inagaki [12] offered a more nuanced split between skills, differentiating between "routine" and "adaptive" expertise. Individuals, who have learned a set of routines that they master and continue to become more efficient in applying, are said to have routine expertise. They will continue to learn throughout their lifetime but will often only apply their new knowledge in a manner that makes the existing procedures/routines more efficient. On the other hand, an individual with adaptive expertise will utilize the knowledge they obtain and apply it to new, innovative procedures or solutions to problems.

The question for our study is the degree to which software development companies value/search out these respective skills sets, and to what extent they report 4-year CS programs and coding boot camps have the capacity to inculcate these crucial elements.

3.1 Participants

Fifteen industry representatives/hiring managers (8 females, 7 males) from twelve different software development companies voluntarily participated in this study. Three of these companies were classified as small sized (<50 employees), four were classified as medium sized (51-250 employees), and five were classified as large sized (>250 employees). Additionally, eleven national boot camp volunteer representatives (4 females, 7 males) from eleven different boot camps participated. The third group of participants consisted of nine university professors (3 females, 6 males) from nine different undergraduate computer science programs.

Participants were recruited through online postings via the National Center for Women in IT (NCWIT), the Special Interest Group in Computer Science listserv (SIGCSE), "cold" emailing Boot Camps directly through the contacts made available through CourseReport [4], as well as "cold" emailing to prospective industry representatives, and through connections via the grant's advisory panel.

3.2 Data Collection & Analysis

From August 2016 through December 2016, the research team conducted a total of three industry focus groups and four one-on-one interviews with industry managers/ directors about their hiring needs and procedures for entry-level software developers. Questions focused on (1) student/ employee recruitment and retention; (2) skills, knowledge, and competencies expected by graduates of each training ground, and (3) what companies expected of code camps and universities alike. All focus groups were semi-structured and intended to provoke free responses and wider discussion. The boot camp and university participants (instructors or faculty who determine their department's CS curriculum) were likewise interviewed with questions focused on (1) curriculum design/ implementation (2) recruitment, admission, and assessment procedures (3) student profiles and (4) skills emphasized within program.

All focus groups and interviews were subsequently transcribed and qualitatively analyzed using both NVivo and Dedoose software. The research team constructed a preliminary coding schema [13], consisting of a series of primary overarching coded categories and two levels of subcategories (i.e. "secondary" and "tertiary" codes).

With their coding, the research team had a threefold purpose. First, to what extent was the drafted coding schema effective as a tool to consistently capture and classify relevant participant responses? Second, to what degree was there inter-coder reliability with regards to the utterances selected to be coded? Campbell [14] refers to this selection of coded utterances as "unitization". Third, to what degree would their codes match on the overarching primary, secondary, and tertiary levels? Reliability was calculated based on three sample coding comparisons between two coders. Across these three, there was 100% agreement on the 4 primary codes, 94% agreement on the secondary codes, and 87% agreement on the tertiary codes. As expected, primary codes saw the highest percentage of agreement, while the increasingly nuanced secondary and tertiary sub-categories saw inter-rater agreement slightly

decrease incrementally. Such a decline is expected as the increased nuance and specificity of the sub-categories can result in a wider fluctuation in interpretation [15]. Table #1 offers the project's overall research questions and coding schema.

Primary code	Secondary (and Tertiary Codes)
RQ: What kinds of learners are attracted to different learning opportunities and why?	
Recruitment and Admission	Program for Recruitment & Admission; Recruitment of a Diverse Student Body; and Student Reason or Motivation
Learner Profile	Demographics; Cognitive Characteristics; Interpersonal Characteristics; and Intrapersonal Characteristics
RQ: What kinds of s/w development learning opportunities are offered by undergrad programs? Boot camps?	
Learning Opportunities	Training Environment (Professor Designed, Student Experience, Underrepresentation, and Institutional Realities); Programmatic Outcome (Anticipated or Stated, and Actual or Experienced); Practice; and Skills and Knowledge (Developing Cognitive Skills, Developing Interpersonal Skills, and Developing Intrapersonal Skills)
RQ: How well do the settings align with regional s/w development industry's needs?	
Industry Needs	Hiring Practices/Standards and Expectations for Hiring; Industry Partnerships; Skills and Knowledge (Cognitive Skills, Interpersonal Skills, and Intrapersonal Skills); and Working Environment (Employee Experienced, Underrepresentation, and Workplace Expectations/Realities)

TABLE #1: Research Questions and Coding Schema

4. RESULTS

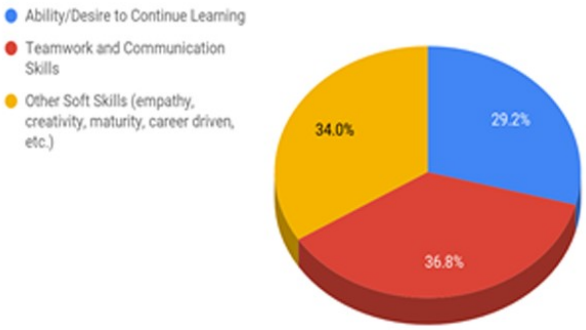
4.1 Soft Skills & Hard Skills as Hiring Indicators

Over the course of the hour-long focus groups with industry representatives, "soft" skills were uniformly the most prevalent topic of discussion. 100% of industry participants noted the importance of soft skills in hiring and as the table below indicates, soft skills were twice as likely to come up than "hard" skills in the focus groups discussions.

	Utterances About Soft Skills	Utterances About Hard Skills	Total Utterances
Small sized companies	67	20	87
Medium sized companies	35	14	49
Large sized companies	83	62	145
Total	185	96	281

TABLE #2: Mention of "Soft" Versus "Hard" Skills Based on Company Size

Multiple industry participants pointed to how extensive their interview process is, as this is where "soft" skills are assessed. The graph below represents those requisite "soft" skills that came up most often among focus group participants, with teamwork and communication (36.8%) as the leading expectation of new hires and with the ability/ desire to continue learning as the second leading expectation (29.2%).



GRAPH #1: Breakdown of Soft Skills desired by Industry Representatives

90% of these industry participants also indicated that routine skill sets, such as the capacity to program, were a necessary initial indicator for hiring. A director of a Southeast big data engineering company noted that "you have to have the technical baseline understanding and aptitude. You wouldn't pass that first screening if you didn't have that." Several of the participants also noted that these requirements often change rapidly within this industry; whereas, the desire to hire individuals, who demonstrate "soft" skills, such as teamwork and the ability/desire to continue to learn, remains consistent with time. In fact, as one hiring manager stated, "if it's someone that they don't think they could get along with, they simply will turn the person down, even if that individual has very strong skills in terms of the technical know-how."

4.2 Employer Preference: Coding Boot Camp or 4-year degrees?

Several companies listed a 4-year degree (or higher) as essential for hire. Many of these companies did not list a specific preference for CS university degrees. In fact, some indicated that hires with alternative degrees, such as music, physics, or education degrees were successful due to similar problem solving traits as CS graduates as well as "the breadth of education that you would get from a 4-year degree, just from baseline knowledge, and experience perspective, versus people that have come through maybe a boot camp situation." A hiring manager from a small data analytics company listed a preference for candidates with a 4-year degree or higher because "some of the practice areas, especially in a data science area, require a pretty heavy mathematical background, so they need an education in statistical math that's deeper than you would get through a boot camp." Three companies recognized

university graduates for having more theoretical knowledge and knowing "the best algorithm to make something work". Some companies do not even bother to interview boot camp graduates due to either having enough college candidates to choose from or not "view[ing] it as much of an asset" to the company.

On the other hand, several companies (including companies focused on consulting, software development and support, automobile technology, and digital marketing for education and healthcare) stated that in certain cases, they actually prefer their boot camp hires. These participants praised their coding boot camp hires based on the fact that they were often older and had more experience with hands-on projects as well as more efficient problem-solving skills based on their work and educational experiences. Furthermore, software development companies prefer candidates with up-to-date knowledge. Four of the managers interviewed noted that a boot camp hire tends to have more current knowledge; whereas, the undergraduate hire may not have been exposed to newer software programs/hardware. Participants also indicated boot camp hire strengths in the areas of communication, code development, more experience pairing, better reactions to critical feedback, and passion and perseverance in work projects. For example, a large online travel and entertainment company states "when you have this input algorithm that works, and then you have someone that's basically just putting it into code, the code you end up with is much more readable and much more logical, and just much more common sense coming out of a boot camp graduate than from a computer science graduate". A large automobile technology company stated that CS graduates are "generally not so much taught to code. They don't come out as the best developers, and they may be able to analyze a problem very well... They can think about the theory really well and explain the theory of things really well, but then when they actually sit down to code, their habits aren't great. Their naming conventions aren't great. They generally are unfamiliar with just basic development tools."

It was also notable that more often than not, code camp hires actually also had a college degree, albeit not in CS. In the focus groups with code camp administrators, we learned that most of their students already had at least a four-year degree (66% on the low end to 80%, on the high end), typically in a liberal arts major/ concentration.

4.3 Perspectives from Coding Boot Camp & Undergraduate CS Programs

As the data in the above sections indicate, positive hire attributes include a mix of both soft and hard skills. Prior work experience counts considerably as does the desire for continuous learning, the capacity to work in teams and ability to receive critical feedback. In terms of hard skills, industry representatives regularly cited the need to know CS principles (data analysis, algorithms) and have a solid background in rigorous mathematics. Industry also pointed toward the need for efficient coding practices and some stressed up to date knowledge of current software development platforms.

So where are these skills inculcated within undergraduate CS programs and coding boot camps? Of the nine CS undergraduate faculty members who participated in this study, all indicated that the development of hard skills is built into their respective curricula. Mandatory classes are offered in data structures, design patterns, software architectures, as well as classes that focus on understanding how computer science is used on the job. Undergraduates instructors were less agreed upon where their students learned soft skills in the classroom, such as teamwork and clear communication. Participants reported that often these skills were not explicitly taught in their classrooms but were tacitly encouraged through the nature of assignments. As one professor points out, simply exposing undergraduates to multiple languages helps them "get over fear of, 'Oh, here's another language I have to learn.'" She added this this exposure to multiple languages also had the side benefit of making her students "more willing to go and explore other languages and take on opportunities." Sometimes instruction was more explicit. Multiple undergraduate instructors reported that they would directly tell students about the importance of continuous learning; some engaged in role playing with senior capstone work, while others had students work in pairs and in teams on select assignments. Six of the nine college/ university participants mentioned inviting in guest industry speakers as a way to ground students' learning in actual application. However, undergraduate CS faculty also reported on the challenges associated with inculcating so-called "soft" skills, such as large class sizes, their own inability to change departmental curricula in a timely manner to meet industry needs, and the challenge of teaching from textbooks in which the solutions to the problems have already been posted on the web. Some professors stated outright that they do not believe that teaching the latest languages and platforms was the role of the university. One professor stated that "the university is about education and not training."

The eleven coding boot camp instructors and administrators, who participated had no such problem categorizing their programs as training. The majority classified their programs as a form of experiential learning and spoke to the daily projects and assignments, which often mimicked real life projects and workplace conditions. As one boot camp representative stated, "the best way to learn is to actually teach. The second-best way to learn is by actually by doing it yourself. Experiential learning…That's a large part of what they do here and that's a large part of what their employer will expect to do as a professional." Other methods used to develop student skills include job prep courses, internships, options to repeat modules following feedback, alumni mentorship programs, seat assignments, project-based learning, portfolio development, development of continuing learning groups for alumni, lectures, group projects, and forming industry partnerships and collaborations. With regards to class size, one boot camp participant noted "Sometimes we have 25 students in the class, sometimes we have more like 15." Finally, coding boot camps reported to be necessarily more responsive to current industry needs. As one administrator stated, code camps are in "a unique

position to pivot really quickly, and address those needs." They are able to change curriculum in between student cohorts in order to" fix things that are not working, get the feedback, improve it and so on."

5. DISCUSSION

Our national focus group discussions with software development managers support the same conclusion that Li, Ko, and Zhu [9] posit in "What Makes a Great Software Engineer?": software development is very much a *socio*technical undertaking, not just a technical one. But based on our data, for industry, this "socio" element takes considerable more effort in terms of recruiting and retaining new talent.

While the research presented here is still in early stages, it does not appear from these focus groups that code camps are a threat to four-year programs. Data in this study in fact suggests that most boot camp hires also have an undergraduate degree. Also, it appears that there are human resource requirements mandating hiring candidates with at least a college degree. However, this data also suggests that companies, which prefer college degrees do not necessarily prefer a CS degree. Furthermore, while our preliminary research with industry managers and directors is hardly a representative sample, it does indicate that some companies are quite willing to interview and hire boot camp graduates—a finding supported by Course Report's [4] annual job placement statistics for boot camp graduates. It is likewise quite clear from our research that the hard (routine) skill of programming is increasingly not enough to land a job. At this point, coding boot camps seem to be fulfilling the role of continuing educational facilities to assist with career growth, career change, and add additional skill sets for resume building. While this may not affect undergraduate CS programs, it could impact future graduate school admissions as candidates may opt for the less expensive route for further education.

What is less clear from our research at this time is to what degree undergraduate programs and coding boot camps inculcate the leading soft and hard skills we have detailed here. Are teamwork and perseverance, in fact, skills that can be developed in these educational programs? Are they even skills, or are they attributes that the individual student enters a program with?

In the next phase of our data collection—individual interviews with CS undergraduates and code camp students—we begin to investigate these questions more directly. What technical proficiencies have these students gained over their respective post-secondary educational environments? What soft skills have they learned (or further developed)? Or do they feel these skills are ones they already had prior to admission? Among those first-year graduates who have already landed a position at a software development company, to what extent did their respective training prepare them for year one of the job? This next step offers a corollary and balance to the industry perspectives we have reported here.

ACKNOWLEDGEMENT

The writing of this paper was supported by through a pair of National Science Foundation (NSF) Core Research and Development Grants to the first and third author. The views expressed in this paper are their own and not necessarily those of the NSF nor their respective institutions.

REFERENCES

[1] The U.S. Bureau of Labor Statistics. 2017. Retrieved from https://www.bls.gov.

[2] Code.org. 2017. Computer science education stats. Retrieved from https://code.org/promote.

[3] Computing Research Association. 2017. Generation CS: CS undergraduate enrollments surge. Retrieved from http://cra.org/data/Generation-CS.

[4] Course Report. 2017. Retrieved from https://www.coursereport.com/.

[5] College Data. 2017. What's the price tag for a college education?" Retrieved from http://www.collegedata.com/cs/content/content_payarticle_tmpl.jhtml?articleId=%2010064

[6] Margo Greenman. 2016. Intensive training helps coding newcomers land technology jobs. (September 2016). Retrieved from http://425business.com/ intensive-training-helps-coders-land-technology-jobs/.

[7] Steve Lohr. 2017. A new kind of tech job emphasizes skills, not a college degree. (June 2017). *The New York Times.* Retrieved from https://www.newyorktimes.com/2017/06/28/technology/tech-jobs-skills-college-degree,html?mcubz=2.

[8] Sarah McBride. 2016. Want a job in Silicon Valley? Keep away from coding schools. (December 2016). *Bloomberg Technology.* Retrieved from https:// www.bloomberg.com/news/features/2016-12-06/want-a-job-in-silicon-valley-keep-away-from-coding-schools.

[9] Paul Luo Li, Andrew J. Ko, & Jiamin Zhu, J. 2016. What makes a great software engineer?" n *Proceedings of the 37th International Conference on Software Engineering - Volume 1* (ICSE '15), Vol. 1. IEEE Press, Piscataway, NJ, USA, 700-710.

[10] James Pellegrino. 2016. What is the evidence on teaching & learning 21st Century skills? *Presentation at the National Research Council on 21st Century Skills.* Retrieved https://www.youtube.com/watch?v=RctULp9Gugo.

[11] James Hiebert & P. Lefevre. 1986. Conceptual and procedural knowledge in mathematics. *Science and Education* (pp. 1-27). Hillsdale, NJ: Erlbaum.

[12] Giyoo Hatano & Kayoko Inagati. 1986. Two courses of expertise. In H. Stevenson, H. Azuma, & K.

[13] Matthew Lombard, Jennifer Snyder-Duch & Cheryl Campanella Bracken. 2010. Practical resources for assessing and reporting intercoder reliability.. Retrieved from http://matthewlombard.com/reliability/.

[14] John L. Campbell. 2013. Coding in-depth semi structured interviews: Problems of unitization and intercoder reliability and agreement. *Sociological Methods & Research ,42 (3),* 294-320.