

Hack.edu: Examining How College Hackathons Are Perceived By Student Attendees and Non-Attendees

Jeremy Warner
UC Berkeley
Berkeley, CA, USA
jeremy.warner@berkeley.edu

Philip J. Guo
UC San Diego
La Jolla, CA, USA
pg@ucsd.edu

ABSTRACT

College hackathons have become popular in the past decade, with tens of thousands of students now participating each year across hundreds of campuses. Since hackathons are informal learning environments where students learn and practice coding without any faculty supervision, they are an important site for computing education researchers to study as a complement to studying formal classroom learning environments. However, despite their popularity, little is known about why students choose to attend these events, what they gain from attending, and conversely, why others choose *not* to attend. This paper presents a mixed methods study that examines student perceptions of college hackathons by focusing on three main questions: 1.) Why are students motivated to attend hackathons? 2.) What kind of learning environment do these events provide? 3.) What factors discourage students from attending? Through semi-structured interviews with six college hackathon attendees (50% female), direct observation at a hackathon, and 256 survey responses from college students (42% female), we discovered that students were motivated to attend for both social and technical reasons, that the format generated excitement and focus, and that learning occurred incidentally, opportunistically, and from peers. Those who chose not to attend or had negative experiences cited discouraging factors such as physical discomfort, lack of substance, an overly competitive climate, an unwelcoming culture, and fears of not having enough prior experience. We conclude by discussing ideas for making college hackathons more broadly inclusive and welcoming in light of our study's findings.

CCS CONCEPTS

• **Computers in Society** → General;

KEYWORDS

college hackathon; informal learning; situated learning

1 INTRODUCTION

A hackathon is an event where people gather in one location to create prototype software projects within a short time period, usually from one day to one week. This term originated in 1999 when

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICER'17, August 18-20, 2017, Tacoma, WA, USA.

© 2017 ACM. 978-1-4503-4968-0/17/08...\$15.00

DOI: <http://dx.doi.org/10.1145/3105726.3106174>

OpenBSD and Sun Microsystems hosted hackathons for developers to create software on their respective platforms [6]. These events have become popular over the past decade and now exist in several forms: 1.) Technology companies host hackathons to promote their APIs [11, 22], 2.) open-source software projects host hackathons to make concentrated bursts of progress [6, 30], 3.) governments and nonprofits host civic hackathons to create technologies for social good [7, 21], and 4.) colleges host hackathons for students [24].

One of the most prominent types of hackathons to arise in recent years is the *college hackathon*, which is usually a 24- to 36-hour event held on a college campus where students create software projects ("hacks") and optionally compete for prizes. Hundreds of students travel to attend large hackathons at schools such as MIT, UPenn, and the University of Michigan, with their bus or plane fares paid by corporate sponsors who view these events as recruiting opportunities. College hackathons started in the U.S. in 2010 [18] and have now become popular enough that there is an organization, Major League Hacking [5], that tracks the status and progress of attendees throughout each "season" (i.e., semester).

In 2016, Major League Hacking sponsored over 200 college hackathons with over 65,000 total participants [5, 24]. We also found that hackathons have occurred at all 47 of the 47 top-ranked computer science departments in the U.S., according to the U.S. News Top 40 rankings [2] (there are 47 since eight schools tied for rank 40). Press articles [24] and personal anecdotes indicate that hackathons are now well-advertised social events in computer science departments, with students viewing them as opportunities for project-based learning, socializing, community-building, and job hunting.

However, despite the rapidly rising popularity of hackathons on college campuses around the world, little is known about why many students choose to attend them, why they find it engaging to spend their weekends coding intensively, what they gain from attending, and, conversely, why other students choose not to attend.

Researchers have recently begun to study corporate [22], scientific [30], and civic [7, 15, 21] hackathons, which are attended mostly by working professionals. Our paper complements this existing literature by presenting one of the first studies of college hackathons. Since these events are popular on-campus venues for informal and situated learning [23] where students learn about programming-related technologies from each other without any faculty supervision, college hackathons are an important yet underexplored site for computing education researchers to study as a complement to studying formal classroom learning environments.

In this paper, we focused our exploratory study on surfacing the perceptions of both students who attended and those who chose not to attend college hackathons. To get a broad range of opinions, we used data from semi-structured interviews with six attendees (3

female, 3 male), direct observation at a hackathon in our university, and 256 survey responses from U.S. college students (42% female).

We found that students were motivated to attend hackathons for both social and technical reasons, that the unique time-limited environment generated excitement and helped them focus intensely, and that learning occurred incidentally, opportunistically, and mostly from peers. Those who chose not to attend or had negative experiences at hackathons mentioned discouraging factors such as physical discomfort, an emphasis on making superficial demos rather than substantive technologies, an overly competitive climate, fears of not having enough prior programming experience to meaningfully contribute, and a sometimes-unwelcoming culture.

We conclude by suggesting ways to make college hackathons more inclusive and welcoming. As hackathons continue moving toward the mainstream of computer science student culture at universities around the world [24], it is important to broaden participation because these are not merely venues for socializing and learning, but are also potential job-seeking opportunities as more companies start recruiting from hackathons in lieu of traditional on-campus career fairs [24]. A lack of inclusion at these events means that certain groups (e.g., women, underrepresented minorities) miss out on opportunities for learning, networking, and jobs.

The contributions of this paper are:

- One of the first studies of college hackathons, which contributes to computing education research by documenting how students informally learn and practice coding at a type of event that is now popular across many campuses.
- Student perceptions of college hackathons, including motivations for attending and not attending, how and what they learned, lasting impacts, and criticisms, based on interviews with 6 students and 256 survey responses.
- Recommendations for improving college hackathons to make them more inclusive and welcoming.

2 BACKGROUND

2.1 Related Work

Despite increasingly frequent mentions of hackathons in popular press articles [11, 12, 24], there have been relatively few academic studies of hackathons. Most of these prior studies have focused on hackathons organized by corporations, nonprofits, and governments rather than those by and for college students. For instance, Komssi et al. performed a case study of five hackathons hosted at an enterprise software company to prototype potential product ideas [22]. They found that these events are effective for getting people from different parts of the company to cross-fertilize expertise with one another. Irani documented her experiences as a participant-observer in a five-day civic hackathon in India to generate ideas for open governance [21]. She found that it infused participants with an energetic, entrepreneurial, action-oriented spirit to solving social problems, spurred by the urgency of a short time limit. Ames et al. studied the use of evangelistic metaphors to encourage volunteer participation in a civic hackathon for disaster relief technologies [7]. Trainer et al. studied team formation, social tie building, and follow-up routines at three scientific hackathons attended by professional scientists and research programmers [30]. They found that different

ways of organizing hackathons (i.e., how teams are formed) resulted in tradeoffs between participants building stronger social ties and more effectively advancing preset technical goals.

In the computing education literature, the few known studies of college hackathons have all been performed by the organizers of these events. For instance, in the StitchFest project [29], researchers created a specialized wearable devices hackathon to broaden participation amongst women in computing, and then studied its 33 attendees. Anslow et al. [8] wrote up an experience report on organizing specialized data science hackathons (“datathons”) for teaching students data science as a supplement to university data science curricula. Nandi and Mandernach [28] organized a series of larger general-purpose hackathons at their university and performed a quantitative analysis of those events by documenting attendee majors, Git source code commit logs, and effects on attendee GPAs.

Unlike these prior studies, to our knowledge, we are the first to investigate college hackathons as third-party outside observers rather than as organizers. We are also the first to study non-participation to understand why students choose not to attend hackathons.

More broadly, our work relates to studies of informal learning of programming. For instance, Dorn and Guzdial studied graphic and web designers learning end-user programming outside of university settings [16, 17]. Boustedt, McCartney, et al. studied CS students’ perceptions of and motivations for self-directed learning [10, 27].

2.2 What are College Hackathons?

Before presenting our study, we first provide background information on what college hackathons are and who attends them.

In the rest of this paper we will use the term “hackathon” to refer exclusively to college hackathons. In addition, note that although student groups also organize smaller niche hackathons for specific purposes (e.g., learning to use a certain technology or focusing on a specific underrepresented group), in this paper we use “hackathon” to refer to the larger-scale general-purpose hackathons that are the most popular instances of this event format.

A hackathon is a 24- to 36-hour free weekend event hosted in a large indoor open space within a college campus. Although these events are organized by students, corporate sponsors pay for food, prizes, and transportation. In return, they get publicity and access to participants for recruiting. At the start of the event, participants mingle, start forming teams, and brainstorm project ideas. Most of the participants’ time is spent coding (“hacking”) to produce prototype apps. There are sometimes technical talks in classrooms, social events such as games, giveaways of free gifts such as company-sponsored water bottles and shirts, and company representatives walking around to mentor or recruit. At the end of the event, teams can optionally present their project to a panel of judges to compete for prizes. Sometimes winners receive immediate internship or job offers at sponsoring companies [24].

Hackathons started in the Northeastern United States in 2010 [18] and spread across U.S. college campuses. Major League Hacking [5] (MLH) tracked data on over 200 hackathons in 2016, and we found that all 47 of the 47 top-ranked U.S. computer science departments [2] have hosted their own hackathons. Many more colleges host smaller hackathons that are not tracked by MLH. Most attract around 100 participants, but the largest can attract well over 1,000.

Hackathons have also spread to other countries [5], but we do not have detailed data on their growth at non-U.S. campuses.

For large general-purpose college hackathons, many participants are undergraduate students who travel to attend via carpool, bus, or plane, with their transportation costs reimbursed by corporate sponsors [18]. According to self-reported interests on participants' Devpost [3] profiles, web programming is currently the most popular interest, followed by mobile programming on Android/iOS.

2.3 Theoretical Framework

The theory of situated learning [23] provides a framework for contextualizing our study. Situated learning posits that people are motivated to learn by doing authentic hands-on activities that bring them closer to joining a desired community of practice. We found that hackathons represent such a type of activity for some students in computer science (CS) and related majors.

Situated learning requires the activity to be *authentic*: learners want to feel like they are working with real tools that professionals in their field use, not just “toy” environments designed for pedagogy [19]. In the context of software-focused hackathons for CS students, this means getting the chance to use programming languages, libraries, frameworks, APIs, and tech platforms that are now the most popular in industry to develop prototypes of industry-relevant projects.

This theory also describes how some people are motivated to learn to join a *community of practice* of professionals in their field and to acquire the values of that community. Many CS students today want to join a community of professional web and mobile application developers at top tech startups and big companies (e.g., Google, Facebook). The presence of professional software engineers attending college hackathons as mentors, guest speakers, and judges gives students an opportunity to interact with and learn from experts in their intended community of practice. In addition, students can learn from more experienced peers on their teams who have done internships at these tech companies.

More broadly, situated learning is loosely related to *social learning theory* [9], which posits that learning occurs within a social context where people directly observe and imitate others around them, often without formal instructional procedures. A hackathon provides one such social context for students to learn technology-related topics from peers.

Finally, although we observed instances of situated learning in our study, we could not see longitudinal effects associated with some aspects of this theory since hackathons are – by definition – short-lived events. For instance, we did not see legitimate peripheral participation, where newcomers gradually move toward the center of a community of practice by taking on core responsibilities over time. Thus, this framework should be used only as an informal lens for our study, but cannot encompass all of our findings.

3 RESEARCH QUESTIONS

Our study examines students' perceptions of college hackathons by focusing on three questions:

- 1.) Why are students motivated to attend college hackathons?
- 2.) What kind of learning environment do these events provide?
- 3.) What factors discourage students from attending?

4 METHODS

To discover students' expectations for and experiences at hackathons, we first performed a qualitative case study by gathering data using semi-structured interviews and direct observations at a college hackathon. Then as a follow-up study, we surfaced the opposite perspective – why students do not like to participate in these hackathons – via an online survey sent to four U.S. universities.

4.1 Case Study of Six Hackathon Attendees

We performed a qualitative case study [31] on students selected from the population of undergraduates at the University of Rochester (a mid-sized Ph.D.-granting U.S. university) who had registered to attend our university's annual hackathon. We emailed the registration list to ask for unpaid volunteers for our study. Then to maximize the diversity of perspectives, we selected six subjects to balance both gender (3 female, 3 male) and amount of prior hackathon experience (3 with no prior experience, 2 with some experience at 2–3 prior hackathons, 1 with lots of experience at 5 prior hackathons). Here are the pseudonyms for the six case study subjects, coded by both gender and ascending experience level:

Pseudonym: # hackathons attended:	Male			Female		
	M1	M2	M3	F1	F2	F3
	0	3	5	0	0	2

All six subjects were undergraduate computer science majors, which is the primary target audience for these kinds of hackathons.

4.2 Interviews and Direct Observations

We conducted three sets of 30-minute interviews with each subject: 1.) one week before our university's hackathon, 2.) one week afterward, and 3.) one month afterward. The first author also observed them working during the hackathon.

The design of our interview questions was initially inspired by prior studies of college hackathons [8, 28, 29] and by the theory of situated learning (Section 2.3). In addition, as each round of interview and observation notes arrived, our research team iteratively coded [14] them to identify common themes, which further refined the list of questions we asked in the next round.

Pre-hackathon expectations interview: This was conducted the week before the hackathon. Its purpose was to assess expectations leading up to the event. We asked the following questions, although each individual conversation was semi-structured and ended up covering different topics:

- What do think the point of hackathons are?
- Why do you want to go to this upcoming hackathon?
- What do you hope to gain from attending the hackathon?
- Do you already have project ideas or team members?
- Who are you hoping to learn from at this event?
- Are you nervous about any aspects of this hackathon?

Direct observations at hackathon: The first author attended our university's hackathon, observed those six students at work throughout the event, and took field notes. This event was a 36-hour hackathon that started on Friday evening at 5pm and ended on Sunday at noon (with 7 hours for sleeping). There were 84 total participants from eight universities, and nearly everyone was an undergraduate

student. The first author struck a balance between being unobtrusive and inquisitive, using their judgment of etiquette from having attended previous hackathons. Since the venue was small (a ground-floor lobby in a single building), the first author was easily able to rotate among the six subjects to observe them at work throughout the course of the event.

Post-hackathon reflection interview: This was conducted the week after the hackathon so that memories were still fresh on subjects' minds. Its purpose was to assess what each subject learned from the event and how their experiences matched prior expectations. Thus, many of our questions followed up on those from the pre-hackathon expectations interview:

- What project did you end up working on?
- Who did you end up learning from?
- What did you end up learning? Anything unexpected?
- What (if anything) has changed about your initial perceptions of hackathons?
- What was most memorable about this hackathon?
- How did attending this hackathon affect your confidence in your coding abilities?

Follow-up lasting-impact interview: This final interview was conducted one month after the hackathon to assess how much of a lasting impact the event had on subjects after they had returned to the routine of school work for a month. We directed this conversation around the following questions:

- What new skills did you learn at the hackathon? (We wanted to see what they remembered one month later.)
- How do you think you will use these skills in the future?
- What criticisms (if any) do you have of hackathons after attending this one? Do you remember any discouraging moments?

4.3 Follow-Up Survey of Hackathon Criticisms

After analyzing data from the six case study subjects, we found their sentiments about hackathons to be mostly positive, since they all chose to attend and were enthusiastic enough to volunteer for our study. Some did express criticism of hackathons, though. Thus, we wanted to dig deeper in that direction by soliciting a broader range of criticisms from students who either chose not to attend these events or who did not have positive experiences there. To achieve this goal, we sent a short survey to undergraduate computer science department mailing lists at four Ph.D.-granting U.S. universities (located in the northeast, south, west, and northwest regions, respectively). Our survey had only three questions:

- If you have never attended a college hackathon, what factors discouraged you from attending? (open-ended)
- If you have attended college hackathons before but did not enjoy the experience, what aspects of the event felt discouraging to you? (open-ended)
- What is your gender? {female, male, other, decline to state}

Since this was a follow-up supplemental survey, we purposely kept it very short and focused to elicit only the information needed to supplement our main interviews and direct observations. We also instructed students to respond only if they were critical of hackathons. We received 256 responses (42% from women).

4.4 Data Collection and Analysis

The first author, who conducted all of the interviews and direct observations, was an undergraduate student who has attended six prior hackathons. While this was beneficial for making subjects feel comfortable talking candidly with someone whom they considered a peer, there is the chance that the first author injected their own biases into field notes. To reduce this risk, we instructed them to write down only raw observations without any interpretation. Then the first author coded the data together side-by-side with a professor who had no prior hackathon experience. Having two coders with varying backgrounds with respect to hackathons provided natural checks and balances throughout the coding process.

Since this was an open-ended qualitative case study, we adopted a grounded approach [14] to code for recurring themes across interviews, direct observations, and survey responses. Two researchers iteratively determined a set of codes together by tallying repeated mentions of themes across these data sources, guided by the frame of situated learning (Section 2.3) and by the first author's personal experiences at hackathons.

5 RESULTS

Based on qualitative data from interviews, observations, and surveys, we present findings on why students are motivated to attend hackathons, in what ways they learn throughout these events, and the main kinds of criticisms from both attendees and non-attendees.

5.1 Mostly-Social Motivations for Attending

We found that students often had social motivations for going to hackathons. The six subjects in our case study primarily viewed our university's hackathon as a weekend social event to attend with their friends and a place to hang out with like-minded people. Many mid-sized to large hackathons like the one we studied are intercollegiate social events where coordinators organize buses to bring students from nearby schools. Students take advantage of this opportunity to travel for free with their friends to see new places and meet new people. M3 summed up his motivations for attending our university's hackathon as: *"You get to be a part of a fun exciting environment, be encouraged to focus intently on a creative solution, meet new people, learn new technology, possibly travel someplace new, and take advantage of company swag [free gift items]."*

In the pre-hackathon interviews, no one felt nervous about finding teammates or a project to work on once they arrived. In fact, M1, M3, and F3 mentioned the social appeal of meeting new people there. Nobody expected to work alone at the event; they knew that hackathons provide group formation activities at the beginning to help attendees find teammates. Prior to attending, M1 and F2 already formed a team with friends from classes, while the other four subjects expected to find teammates there.

Only the most experienced subject mentioned being excited about technical opportunities in addition to social ones. M3, who had attended 5 prior hackathons, said that he was excited to gain access to software development kits, APIs, and hardware provided by sponsoring companies, as well as getting a chance to chat with employees from those companies who attend as mentors. Students could borrow a variety of hardware for the duration of the event. For instance, our university's hackathon featured Pebble smartwatches, 3D printers, Myo gesture control armbands, and Arduinos.

5.2 Situated and Social Learning

Situated and social learning (Section 2.3) emerged as major themes. Attendees had the chance to learn via working on software projects alongside their peers in an informal co-located setting without faculty supervision. In our post-hackathon interviews, subjects' memories about what software prototypes they built at the event were not nearly as salient as what and how they learned there. We classified subjects' recollections about learning into three categories: as being incidental, opportunistic, and from peers.

Incidental: Subjects mentioned learning technical skills at the hackathon as a side effect of trying to get their projects into a properly working state. During pre-hackathon interviews, only F1 mentioned that her goal was to attend to learn a specific skill – web programming from Codecademy (codecademy.com) tutorials – which she ended up doing. Everyone else focused their pre-interview conversation on what they wanted to *do* at the hackathon rather than what they wanted to explicitly learn. Yet at the event, we often observed them learning as a byproduct of doing. Pervasive examples of incidental learning included participants learning how to use software tools such as the Git version control system, how to upload their code to GitHub, how to share files with their teammates using Dropbox, how to deploy code to servers, and how to write Bash scripts to automate command-line tasks. Since these tools helped teams become more productive, members taught one another how to use them on-demand out of necessity as they encountered roadblocks in their projects.

Opportunistic: Attendees also learned opportunistically – taking advantage of short-lived ephemeral opportunities at the hackathon to direct their learning toward sometimes unexpected paths. For example, M3 started the event by working on a web programming project (an RSS feed generator) but saw that Pebble smartwatches were available to borrow at the hackathon venue. He then switched his project to hacking on a smartwatch app, opportunistically learning the Pebble API by reading online tutorials. He enjoyed this impromptu detour, but during the post-hackathon interview, he mentioned that one downside of this ad-hoc approach is that he would not get to reinforce those skills later since he did not own a Pebble smartwatch himself.

From peers: During pre-hackathon interviews, subjects described their perceptions of how learning occurs at these venues, and the theme of peers came up several times. For instance, M1 said that the “learning style consists of goal-oriented peer learning.” M3 said that “hackathons specialize in small group learning, and you feed off of your teammates' energies and enthusiasm.” And F3 said, “hackathons are peer based, [with a] much more democratic learning style.”

Hackathons foster peer-based learning by having students work in teams; many do not allow individual projects to be submitted for judging. Although several mentors from companies and student hacking groups were present at our university's hackathon, there were not enough to help everyone. Less experienced attendees were especially reluctant to approach mentors for help, so they ended up learning more from peers.

Even the attendees who worked alone benefited from the presence of peers. For instance, F1 worked by herself on basic web programming tutorials from Codecademy. However, she felt comfortable asking for help from peers around her. She said that since

the hackathon provided a social context where everyone was doing open-ended technical work, it did not feel awkward to approach people to ask questions about programming. In contrast, she said she would be reluctant to suddenly strike up a conversation about programming while socializing on campus; even if students were working on class assignments together, she felt it would be hard to get them to take a break to help her with unrelated topics.

During post-hackathon interviews, several subjects mentioned peer learning as a memorable part of the event. F2 reported that she learned how to work better in groups, how to split up work, and how to integrate different components under intense time pressure. She also did lots of pair programming to see how others approached the problem and learned by observing more experienced peers. F1, a first-time hackathon attendee, said she looked forward to attending more hackathons next semester based on this experience since she had fun learning while “goofing off” with peers at the same time, which was made possible by the lack of authority figures present.

5.3 Authenticity of Hackathons Versus Classes

During post-hackathon interviews, some subjects reflected on the differences between the learning environments in hackathons and in formal university classes. One emergent theme related to our lens of situated learning (Section 2.3) was *authenticity* – subjects perceived the hackathon environment to be more authentic than university classes in emulating a real-world tech company workplace. They enjoyed the chance to create projects at the hackathon using the latest industry-standard technologies, often provided by sponsoring companies and personally introduced to them by employees from those companies who attended as mentors. In contrast, they said how course curricula created by university professors are usually slower to update to the latest programming languages, libraries, frameworks, and APIs popular in industry today. M3 also mentioned the fact that “you can actually get HIRED [by a sponsoring company] from doing well at a hackathon, which you can't get in the classroom.”

Subjects also mentioned how, in stark contrast to classes, students fully determine what and how they learn at a hackathon. They pointed out that professors are not present, which again simulates a more authentic work environment. F1 summarized the differences as: “[at a hackathon] learning by example and experimentation instead of top down lecture style. Very hands on.” F3 said: “Hackathons are very open ended, you are simply given space, resources, mentors, as well as access to peer mentors, and then encouraged to do something you think is awesome.”

However, they mentioned that the downside was that there is no formal structure or pedagogy at a hackathon, which can be detrimental to effective learning. M3 said that, unlike classes, there is no time for reflection and expert feedback in a hackathon, since there is no professional instructor whose job it is to give feedback to students. He continued, “The main source of feedback is the current functionality of your project, and your peers' perceptions of it.”

Another difference between working on projects at a hackathon and in a class is the time dimension: Rather than lasting for several months, a hackathon lasts for 36 hours at most. Thus, students must pick up and apply newly-learned knowledge immediately rather than waiting for the next assignment or exam to get feedback.

F2 summarized the differences in formats as: “A hackathon provides more creative freedom with projects, shorter time period than classes, and less concern about learning material deeply. I just want to focus on getting [projects] working. There is also no grade, so you don’t face the academic pressure of the classroom. At a hackathon, people will be willing to collaborate and share what they are working on more since they are not working on the same project or for grades.”

Regarding the lack of grades, M1 pointed out that at a hackathon, one can experience “failure without actual failure,” so attendees can take on riskier projects without fears of receiving a failing grade if their attempts do not succeed.

Finally, some attendees bridged hackathons and classes by using their time at the hackathon to work on class projects. For example, M1 went to the hackathon with a friend and planned to use that time to make progress on a class project. Since it was his first hackathon, he was apprehensive about not being able to meaningfully contribute to a team there, but he still wanted to attend to experience the event.

Related to our lens of situated learning, we observed how these students’ desire to join a community of practice of professional programmers motivates them to seek out *authenticity* in their university CS education [19]. They want to learn the latest programming languages, tools, APIs, and frameworks that industry practitioners use. They also reported that, in contrast to many computer science courses taught by professors, which are usually more theoretical or conceptual, hackathons offer a concentrated time and space where they could learn these more practical topics from their peers and from industry practitioners. Although professors are de facto authority figures in university classroom settings, the majority of students do not actually aspire to join the community of practice of college professors. Instead, their role models are older students who have obtained internships at top companies and current employees at those companies, who are often present at hackathons to show them the more pragmatic aspects of software development.

5.4 Lasting Impacts of Attending

During our follow-up interviews (one month after the hackathon) we asked subjects about what knowledge they retained from the event. Subjects could not vividly recall any of the specific skills they learned. Many remembered the event as a blur, with only vague recollections of the general kinds of topics they had learned. They attributed this lack of retention to the learning being *incidental* – they were focused on immersing themselves in the 36-hour experience and had no set curriculum, exams, or follow-up lessons to reinforce their knowledge like they would have in a class. One exception was F3, who purposely set out to learn the Scala programming language during her hackathon project because she knew that she would use it at her upcoming internship.

The most salient lasting impact was not about specific technical skills but rather on students’ improved confidence about their own coding abilities. First-time attendee F1 said that “going to this hackathon improved my self-confidence. I know a lot more than I thought I knew, and feel more normalcy with respect to peers.” F2 reported feeling more confident about working with others on a team. F3 felt that the environment was helpful and supportive for making progress on her technical interests and improving self-efficacy. We

a.) “If you have attended hackathons before but did not enjoy the experience, what aspects of the event felt discouraging to you?”

Total: N=126	Male: N=81	Female: N=43
Discomfort (35%)	Discomfort (33%)	Discomfort (40%)
Novice fears (28%)	Novice fears (22%)	Novice fears (37%)
No team/idea (21%)	No substance (20%)	No team/idea (33%)*
No substance (15%)	Competitive (20%)*	Hacker culture (16%)*
Competitive (14%)	No team/idea (15%)*	No substance (7%)
No time (10%)	No time (11%)	No time (7%)
Hacker culture (8%)	Hacker culture (2%)*	Competitive (5%)*

b.) “If you have never attended a college hackathon, what factors discouraged you from attending?”

Total: N=130	Male: N=61	Female: N=65
Novice fears (48%)	No time (51%)	Novice fears (65%)*
No time (43%)	Novice fears (33%)*	No time (35%)
No team/idea (22%)	No team/idea (18%)	No team/idea (26%)
Discomfort (11%)	Discomfort (8%)	Discomfort (14%)
Hacker culture (5%)	Competitive (3%)	Hacker culture (8%)
Competitive (5%)	Hacker culture (2%)	Competitive (6%)
No substance (2%)	No substance (2%)	No substance (2%)

Table 1: Summary of responses to both survey questions. Each column adds up to more than 100% since some responses contained more than one category. Chi-square statistically significant differences between genders for each category at $p < .05$ marked as * (e.g., Competitive: 20% vs. 5%). Six respondents identified as neither male nor female.

noticed that only the three female subjects reported this effect, but our sample size is far too small to make any meaningful generalizations. Based on these observations, though, we believe that the effect of hackathon attendance on CS student self-efficacy would be an interesting direction for future work.

5.5 Attendees’ Criticisms of Hackathons

The mostly-positive sentiments reported by the six case study subjects prompted us to dig deeper to uncover the opposite perspective: students’ criticisms of hackathons. From analyzing responses to a follow-up survey we sent to four U.S. universities, we grouped attendees’ criticisms into seven categories. Table 1a shows how frequently each category was mentioned in the 126 survey responses to the question: “If you have attended hackathons before but did not enjoy the experience, what aspects of the event felt discouraging to you?” We present them here in descending order of prevalence:

Discomfort: The most common criticism amongst attendees was physical discomfort. Students cited lack of sleep, bad sleeping accommodations on gymnasium floors or dorm couches, unhealthy free food, and lack of personal hygiene of some participants as being uncomfortable. Also, they experienced discomfort and stress due to the time pressures of working intensely over an entire weekend and the loud ambiance of hundreds of students in a crowded space.

Novice fears: The next most common criticism was about the lack of support for novices. Some novices felt uncomfortable about not having enough technical skills to contribute meaningfully to

projects and feared becoming “dead weight” on their team. They also mentioned how hackathons were not the ideal learning environment for novices, since they provide no pedagogical structure. There are technical talks, but those are usually for advertising a sponsoring company’s products or APIs, not to guide novices step by step like a formal class would do. Mentors are practitioners and not trained educators, so they might not be good at explaining fundamental CS1-type concepts. Asking for help can be intimidating, especially as a novice in a room filled with more experienced hackers busy working on their projects.

No team or idea: Attendees who did not come to the hackathon already with a team or idea often had a hard time finding teams. Women reported this problem over twice as frequently as men (33% vs. 15%, chi-square test $\chi^2(1, N = 126) = 4.3, p = .038$). Teams also dissolved mid-way through the event due to ideas fizzling out, team members being flaky, or members not getting along. One survey respondent wrote: “Going to Hackathons with students from [my university] and teaming up with them has always been discouraging as they would just give up on the project half way through and rather focus on going around the town.”

No substance: 15% of respondents noted how hackathons incentivized building flashy app demos rather than creating something more substantive and longer-lasting using computer science principles, since judges (often from companies) are more impressed by cool-looking demos. One wrote, “because there are so many corporate sponsors, students tend to be dissuaded from building meaningful products and instead build toward whichever sponsor has the most attractive prize.” Also, some mentioned how hackathons do not teach rigorous software engineering practices; there are no instructors to critique the technical architecture of one’s project, or how maintainable and robust the code is.

Overly competitive: Although submitting projects for judging is optional, some participants felt like the presence of cash prizes and company recruiters made the atmosphere overly competitive: e.g., “The emphasis on huge prizes lessens the joy I get from hacking. I’d rather hack at a small, no frills, no prizes hackathon with close friends than go to a huge competition where I feel like I’m part of a herd of cattle whose job is to listen to propaganda from sponsors.”

Table 1a shows that men mentioned the overly competitive problem four times more often than women did (20% of men versus 5% of women, chi-square test $\chi^2(1, N = 126) = 4.0, p = .045$). One possible explanation is that more male participants viewed hackathons as a coding contest and cared more about winning, rather than viewing them as recreational social events.

No time: 10% of respondents regretted giving up an entire weekend to attend a hackathon, which is time that they could have spent catching up on sleep, attending social activities, and making progress on homework. One respondent wrote, “Giving up an entire weekend to travel and work was tiring, making it tough to justify going to more in the future given my other commitments, both academic and otherwise.”

Hacker culture: 16% of female attendees cited the presence of “hacker culture” [26] or a “hardcore” ethos [20] as discouraging and sometimes even hostile; only 2% of male attendees mentioned this problem ($\chi^2(1, N = 126) = 6.0, p = .01$). One woman wrote, “Arrogant students, mostly male, who act haughty and patronizing because

I don’t know some obscure JavaScript framework. I feel alienated by these types of people, and also alienated by the images of The Hacker stereotype these Hackathon marketing teams perpetuate.”

5.6 Why Some Students Choose Not to Attend

Since hackathons are now becoming prevalent across college campuses, we wanted to also investigate why students who know about these events choose *not* to attend. Table 1b summarizes 130 survey responses we received to the question: “If you have never attended a college hackathon, what factors discouraged you from attending?” Note that these respondents all knew about hackathons; otherwise they would not have written responses about discouraging factors.

The most common discouraging factor was “novice fears,” with women twice as likely to mention it as men (65% vs. 33%, $\chi^2(1, N = 130) = 11.5, p = .001$). For instance, one woman wrote, “Being afraid that my skills are not enough to actually accomplish anything or that, if I do ask for help, I would be seen as incompetent or bad at programming.” This could be a manifestation of the well-documented gender confidence gap amongst computer science students [20].

The rest of Table 1b shows that even though these respondents had never attended hackathons, they still mentioned all of the same discouraging factors as the attendees did. One possible explanation is that these students heard about what happens at hackathons from peers who have attended, which led them to form those same negative preconceptions that discourage them from attending.

However, note that this is still a self-selected sample of non-attendee respondents; many other college students do not attend hackathons likely because they do not know much about them, or they do not have strong enough opinions to report in a survey.

6 STUDY LIMITATIONS

We performed a case study on six students who all attended a hackathon and who were willing to undergo three interviews. This volunteer sample may be biased toward more social or autodidactic students. To compensate, we obtained a broader set of perspectives by surveying 256 CS students at four U.S. universities. As with any case study, we cannot guarantee that its subjects are representative of all college hackathon attendees, but we strove to achieve some level of diversity along two dimensions: gender and amount of prior hackathon experience. Also, our findings may not generalize to smaller niche hackathons, those with specialized themes, or those that do not involve corporate sponsors.

Additionally, since the focus of this study was not on detailed demographic effects, we collected some data on gender but did not investigate race, ethnicity, or socioeconomic status. All case study subjects were either white or Asian. We also did not investigate the possible effects of intersectionality [13] (e.g., the experiences of women who are also underrepresented minorities). All data were collected from four-year Ph.D.-granting U.S. universities, so other types of higher education institutions were not represented.

Finally, our study is exploratory and high-level in nature because it is one of the first to investigate general college hackathons. Thus, we did not focus on the challenges faced by specific groups of students (e.g., women, underrepresented minorities, first-generation college students) in detail. Targeted follow-up studies are needed to uncover the deeper and richer nuances behind their experiences.

7 DISCUSSION

Relationship to prior work: As the first known study of college hackathons performed by outside researchers rather than by the organizers themselves, our findings about the energy and enthusiasm generated by these events, along with their abundant informal learning opportunities, corroborate similar findings from past hackathon organizers [8, 28, 29]. We deepened these prior results by making ties to situated learning theory, especially highlighting attendees’ desires to join a community of practice of professional programmers and perceiving hackathons as more authentic learning environments than classes (Section 5.3). We also augmented prior studies with new perspectives from hundreds of students who either did not enjoy or did not attend hackathons (Table 1).

Implications for teaching: An ongoing challenge in computing education is how to balance teaching the foundational theoretical concepts of computer science with the pragmatic (but fast-changing) programming skills that students want to learn to get jobs. How can some positive elements of hackathons be infused into traditional university classes? One idea is for classes to hold mini-hack-days for students to implement small open-ended projects using concepts from those classes. Case study subject F1 mentioned that a *“hackathon is a time for implementation of theoretical concepts learned in class, so professors should support this type of thing.”* M1 pointed out the importance of experiencing *“failure without actual failure”*, so hack-day assessments should be formative and not high-stakes. Another advantage of adapting hackathons into existing classes is the possibility of follow-up feedback and iteration after the events are over, which is lacking in current hackathons.

Implications for computing education research: Our high-level exploratory study only scratches the surface on investigating the recent phenomenon of college hackathons, so follow-up computing education research is needed to hone in on questions such as: What are the longer-lasting impacts of hackathons on students’ self-efficacy and future job success? How do hackathons relate to the rich ecosystem of other informal computing learning environments on campuses such as student-run makerspaces and hobby computing clubs? How effective are hackathons at fostering rigorous learning and retention rather than simply serving a sociotechnical purpose? What randomized controlled experiments can we perform at hackathons to isolate the efficacy of specific event components?

Making hackathons more inclusive and welcoming: Given their recent popularity, it is critical to take steps to make hackathons more inclusive and welcoming. Although these have always been desirable goals for any computing-related event, they are now becoming urgent as hackathons move toward mainstream CS student culture. Students now view them not only as social and learning opportunities, but also as venues to obtain coveted job offers in lieu of attending career fairs [24]. As companies are recruiting more out of hackathons, lack of inclusion at these events means that certain groups (e.g., women, underrepresented minorities) are at more of a disadvantage when job hunting. It is no longer a matter of having fun at a social gathering; jobs are at stake.

The challenges faced by female computer science students have been well-documented [20, 26]. Our case study subjects and survey respondents corroborated these prior findings with their concerns

that hackathons embody a type of geeky environment that implicitly excludes women and underrepresented minorities. In our survey, “hacker culture” was cited far more often by women than men as discouraging them from attending. Subjects F1 and F3 suggested that holding women-only hackathons and having more female mentors at existing hackathons would help, although those would only be first steps toward making them feel more welcoming. Another idea was for hackathons to follow the example of diversity-focused industry technology conferences by implementing codes of conduct [1, 4] with zero-tolerance policies for exclusionary behavior.

Making hackathons more generally welcoming for novices can also broaden participation, since students from underrepresented groups likely come into college with less prior programming experience [25, 26, 29]. In our survey, both men and women cited “novice fears” as a discouraging factor, although women were twice as likely to mention it. One way to make these events more novice-friendly is to add lightweight pedagogical structure such as distinct phases and checkpoints, with expert feedback given along the way. Hackathons are now largely unstructured, so novices can easily feel lost. However, adding such structure requires trained mentors to attend. One idea is for organizers to recruit from their school’s computer science TA (teaching assistant) and LA (lab assistant) populations. Many current mentors are software developers from sponsoring companies who are not trained as educators and who have implicit incentives to advertise their own company’s products rather than helping students learn generalizable knowledge.

Finally, as college hackathons grow more popular, they attract more corporate sponsorship. And as companies offer larger prizes and greater prospects of job offers, some students in our survey felt that hackathons are losing their original spirit of making creative technological hacks and are turning into full-fledged competitions. Our survey respondents cited “too competitive” and “no substance” (i.e., making flashy demos just to impress judges) as discouraging factors. We acknowledge that it can be hard for organizers to resist the allure of prestigious companies offering increasing amounts of funding to grow these events, but it is also critical to keep an eye on the core values – collaboration, informal learning, community building – that benefit students and not simply sponsors. Scaling gracefully is an ongoing challenge: How can organizers preserve the maker ethos of small-scale events while growing to hundreds of participants? They could potentially limit sponsor involvement, use funding to pay instructors to come as mentors rather than company representatives, and put modest caps on prizes.

8 CONCLUSION AND FUTURE DIRECTIONS

We have presented one of the first academic studies of college hackathons, which documents student perceptions of these now-popular events from both sides: those who participated and those who chose *not* to participate. Given the growing popularity of hackathons, how should computing educators view them – as threats to traditional CS education or as exciting new opportunities for combining the theoretical with the practical in a situated learning environment? If hackathons are indeed here to stay, then how can we make them more inclusive and welcoming so that more students benefit? And how can they potentially influence traditional classroom teaching? We hope this paper sparks discussion of these questions and more.

REFERENCES

- [1] 2013. PyCon Code of Conduct for the Python community conference. <https://github.com/python/pycon-code-of-conduct>. (March 2013). Accessed: 2016-04-01.
- [2] 2014. U.S. News and World Report: Computer Science Rankings. <http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-science-schools/computer-science-rankings>. (2014). Accessed: 2016-04-01.
- [3] 2015. Devpost - Hackathons, App Contests, and Hackers. <http://devpost.com/>. (2015). Accessed: 2016-04-01.
- [4] 2015. Google Groups: A Code of Conduct for the Go community. <https://goo.gl/RsJsRf>. (June 2015). Accessed: 2016-04-01.
- [5] 2016. About Major League Hacking. <https://mlh.io/about>. (2016). Accessed: 2016-04-01.
- [6] 2016. OpenBSD hackathons. openbsd.org/hackathons.html. (April 2016). Accessed: 2016-04-01.
- [7] Morgan G. Ames, Daniela K. Rosner, and Ingrid Erickson. 2015. Worship, Faith, and Evangelism: Religion As an Ideological Lens for Engineering Worlds. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. 69–81.
- [8] Craig Anslow, John Brosz, Frank Maurer, and Mike Boyes. 2016. Datathons: An Experience Report of Data Hackathons for Data Science Education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 615–620. DOI: <https://doi.org/10.1145/2839509.2844568>
- [9] Albert Bandura. 1977. *Social Learning Theory*. Prentice Hall.
- [10] Jonas Boustedt, Anna Eckerdal, Robert McCartney, Kate Sanders, Lynda Thomas, and Carol Zander. 2011. Students' Perceptions of the Differences Between Formal and Informal Learning. In *Proceedings of the Seventh International Workshop on Computing Education Research (ICER '11)*. ACM, New York, NY, USA, 61–68. DOI: <https://doi.org/10.1145/2016911.2016926>
- [11] Alexandra Chang. 2012. Deep inside a Facebook hackathon, where the future of social media begins. *Wired* (July 2012).
- [12] Lizette Chapman. 2017. These Hackathon Hustlers Make Their Living From Corporate Coding Contests. *Bloomberg Technology* (April 2017).
- [13] Patricia H. Collins. 2015. Intersectionality's Definitional Dilemmas. *Annual Review of Sociology* 41, 1 (2015), 1–20.
- [14] Juliet M. Corbin and Anselm L. Strauss. 2008. *Basics of qualitative research: techniques and procedures for developing grounded theory*. SAGE Publications, Inc.
- [15] Carl DiSalvo, Melissa Gregg, and Thomas Lodato. 2014. Building Belonging. *interactions* 21, 4 (July 2014), 58–61.
- [16] Brian Dorn and Mark Guzdial. 2006. Graphic Designers Who Program As Informal Computer Science Learners. In *Proceedings of the Second International Workshop on Computing Education Research (ICER '06)*. ACM, New York, NY, USA, 127–134. DOI: <https://doi.org/10.1145/1151588.1151608>
- [17] Brian Dorn and Mark Guzdial. 2010. Discovering Computing: Perspectives of Web Designers. In *Proceedings of the Sixth International Workshop on Computing Education Research (ICER '10)*. ACM, New York, NY, USA, 23–30. DOI: <https://doi.org/10.1145/1839594.1839600>
- [18] Jon Gottfried. 2014. A brief history of hackathons (lecture video). <http://youtube.com/Zr6VPAe9CKU>. (March 2014). Accessed: 2016-04-01.
- [19] Mark Guzdial. 2015. Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics* 8, 6 (2015), 1–165.
- [20] Lilly Irani. 2004. Understanding Gender and Confidence in CS Course Culture. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04)*. ACM, New York, NY, USA.
- [21] Lilly Irani. 2015. Hackathons and the Making of Entrepreneurial Citizenship. *Science, Technology & Human Values* (2015).
- [22] Marko Komssi, Danielle Pichlis, Mikko Raatikainen, Klas Kindström, and Janne Järvinen. 2014. What are hackathons for? *IEEE Software* PP, 99 (May 2014).
- [23] J. Lave and E. Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
- [24] Steven Leckart. 2015. The Hackathon Fast Track, From Campus to Silicon Valley. *New York Times* (April 2015).
- [25] Jane Margolis. 2008. *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press.
- [26] Jane Margolis and Allan Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. MIT Press.
- [27] Robert McCartney, Jonas Boustedt, Anna Eckerdal, Kate Sanders, Lynda Thomas, and Carol Zander. 2016. Why Computing Students Learn on Their Own: Motivation for Self-Directed Learning of Computing. *Trans. Comput. Educ.* 16, 1, Article 2 (Jan. 2016), 18 pages. DOI: <https://doi.org/10.1145/2747008>
- [28] Arnab Nandi and Meris Mandernach. 2016. Hackathons As an Informal Learning Platform. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 346–351. DOI: <https://doi.org/10.1145/2839509.2844590>
- [29] Gabriela T. Richard, Yasmin B. Kafai, Barrie Adleberg, and Orkan Telhan. 2015. StitchFest: Diversifying a College Hackathon to Broaden Participation and Perceptions in Computing. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. 114–119.
- [30] Erik H. Trainer, Arun Kalyanasundaram, Chalalaf Chaihirunkarn, and James D. Herbsleb. 2016. How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '16)*.
- [31] Robert K. Yin. 1989. *Case Study Research: Design and Methods*. SAGE Publications, Inc.