Hack.edu: How Collegiate Hackathons Are Perceived By Student Participants and Non-Participants

anonymized for submission

ABSTRACT

Collegiate hackathons have recently grown popular, with tens of thousands of students now participating across hundreds of college campuses. These are 24- to 36-hour gatherings where students collaborate to create original software projects ("hacks") out of existing components. Despite the rising popularity of these events, little is known about why some students choose to attend, what they gain from attending, and conversely, why others choose not to attend. This paper presents one of the first empirical studies of collegiate hackathons, which contributes to the HCI literature by documenting how young people interact with technology and with each other in a unique venue concentrated in both space and time. Through semi-structured interviews with ten participants (50% female), direct observation at a hackathon, and 256 survey responses from college students (33% female), we discovered that students were motivated to attend for both social and technical reasons, that the venue 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, fears of not having enough experience, and hostile hacker culture. We conclude by suggesting ways to make future collegiate hackathons more inclusive.

Author Keywords

hackathon, collegiate hackathon

ACM Classification Keywords

K.4.0. Computers in Society: general

INTRODUCTION

A hackathon is an event where people gather together in one location to create prototype software projects within a short time period, usually from one day to one week. This term was first coined in 1999 when OpenBSD and Sun Microsystems hosted hackathons for developers to create software on their respective platforms [10]. These events started rapidly rising in popularity around 2011 (Figure 1) and now exist in several forms: 1.) Technology companies host hackathons to promote their APIs [14], 2.) open-source software projects host hackathons (also called "code sprints") to make concentrated bursts of progress [10], 3.) governments and nonprofits host civic hackathons to prototype technologies for social good [5, 11], and 4.) universities host hackathons for students.

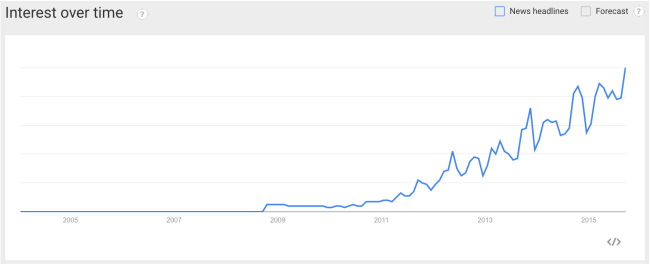


Figure 1. Google Trends [4] shows the popularity of the search term "hackathon" over the past decade (2005–2015), with a rapid rise starting in 2011. The y-axis shows the number of searches for "hackathon" relative to the total number of Google searches in each time period.

One of the most prominent types of hackathons to arise in recent years is the *collegiate hackathon*, which is a 24- to 36-hour event 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 them as recruiting opportunities. Collegiate hackathons started in the U.S. in 2010 [10] and have now grown popular enough that there is an organization, Major League Hacking [2], that tracks the progress of attendees throughout each "season" (i.e., semester). The most devoted students travel to attend a different hackathon almost every weekend.

Major League Hacking sponsored over 150 collegiate hackathons in 2015, with over 50,000 total participants [2, 15]. Hackathons have already occurred at 45 out of the 47 top-ranked computer science departments in the United States, according to the U.S. News and World Report Top 40 [1] (several schools are tied for some ranks). Popular press articles [15] and personal anecdotes indicate that hackathons are now well-advertised social events in computer science departments, with students viewing them as opportunities for hands-on project-based learning, socializing, community-building, and even job seeking (due to the frequent presence of engineers and recruiters from sponsoring companies).

Collegiate hackathons are rich sites for studying HCI and CSCW since they are venues concentrated in space and time where students gather to collaboratively create new technologies out of existing components, opportunistically harnessing whatever tools and learning resources are available on the

Submitted to CHI 2016

spot. However, despite the rising popularity of these events across college campuses, little is known about why some students choose to attend, why they find it engaging to spend their weekends coding with little sleep, what they gain from attending, and, conversely, why other students choose *not* to attend even though they know about these events. In contrast to Figure 1, a Google Scholar search for "hackathon" yields almost no academic papers. The closest related work are HCI and CSCW studies of hackerspaces and maker culture [6, 9, 13, 16, 19, 20, 21]. Researchers have started investigating corporate [14] and civic [5, 8, 11] hackathons attended by working adults, and our work extends this growing body of literature with one of the first studies of collegiate hackathons.

We focused our study on perceptions of college students who both participated in and chose not to participate in collegiate hackathons. To get a broad range of opinions, we used semi-structured interviews with ten participants (5 female, 5 male), direct observation at a hackathon in our university, and 256 survey responses from college students (33% female) interning as software developers at a major technology company.

We discovered that students were motivated to attend hackathons for both social and technical reasons, that the unique environment of being concentrated in time and space 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 flashy demos over substantive technologies, an overly competitive climate (especially at larger hackathons with more prize money), fears of not having enough prior programming experience to meaningfully contribute, and hostile hacker culture (far more frequently cited by female students).

We conclude by suggesting ways to make future collegiate hackathons more inclusive and welcoming. As hackathons move toward the mainstream of computer science student culture at universities around the world [15], it is important to broaden participation because these are not only 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 [15]. A lack of inclusion at these events means that certain groups (e.g., women, underrepresented minorities) miss out on opportunities for informal learning and job-seeking.

The contributions of this paper are:

- One of the first academic studies of collegiate hackathons, which contributes to HCI literature by documenting how young people interact with technology and with each other at a kind of event that is growing in popularity across college campuses, now with tens of thousands of participants.
- Findings about student perceptions of collegiate hackathons, including motivations for attending, how and what they learned, lasting impacts of the experience, and criticisms from 107 attendees and 159 non-attendees.
- Recommendations for improving collegiate hackathons to make them more inclusive and pedagogically meaningful.

BACKGROUND

Although there has not yet been much academic research on hackathons, the related topics of hackerspaces and maker culture have been well-studied by HCI and CSCW researchers.

Hackerspaces are physical sites that encourage technical innovation via tinkering and ad-hoc tool-making [6, 16], community-building via DIY (do-it-yourself) maker culture that exists outside of formal professional settings [9, 20, 21], and informal learning both inside and outside of school settings [13, 19]. A subset of research into hackerspaces has investigated how they foster the values of specific demographics of hackers: For instance, Fox et al. focused on how feminist hackerspaces provide safe, supportive environments to build the professional and personal identities of female hackers [9]. Sun et al. identified the strong influence of cultural context on the values of niche DIY communities, using the example of how elderly electronic hackers in modern China are motivated by their childhood memories of 1950s-era China [20]. Our study focuses on the hacker demographic of computer science college students at four-year U.S. universities in the mid-2010s, where one trend is to value creating Web and mobile apps to show to peers for recognition and to technology company representatives for the chance to obtain internships and jobs [15].

Although both hackerspaces and hackathons attract participants who share similar values (e.g., creating technology by remixing existing components, small-group collaboration, hands-on learning), the two differ in several ways. The most salient is in the time dimension: Hackathons last for one or a few days, while hackerspaces last for months or years. Also, due to their existence as persistent physical spaces, hackerspaces support more diverse kinds of projects such as those in electronics, crafting, mechanical engineering, furniture making, and even food hacking. In contrast, many hackathons focus on software development due to the relative ease of finishing a software demo in 24 to 36 hours. Finally, many hackerspace participants are working adults [9] or retirees [20] with different preferences (e.g., not wanting to pull all-nighters) than the students we study in collegiate hackathons. One notable exception is the Hackademia [13] project, which brings hackerspaces into a university to promote technical literacy amongst non-STEM students.

The few prior academic studies of hackathons have focused on those 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 [14]. They found that corporate hackathons are good for testing out ideas that might someday turn into products and for getting people who work in different parts of the company to socialize and cross-fertilize expertise with one another. Irani documented her experiences as a participant-observer in a five-day hackathon in India to generate ideas for open governance [11]. She found that these hackathons infuse participants with an energetic, entrepreneurial, action-oriented spirit to solving social problems, spurred by the urgency of producing a demo within a

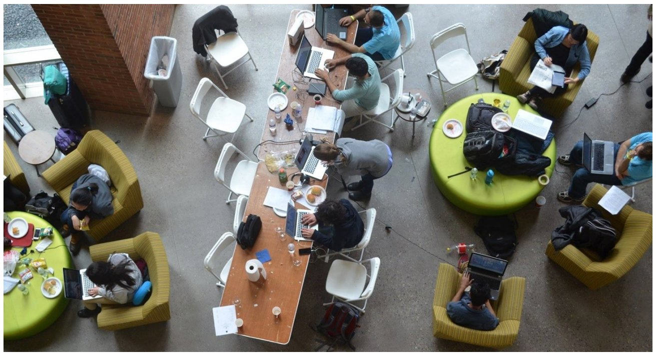


Figure 2. A typical collegiate hackathon venue (at our own university), with students working together on laptops in a large indoor open space.

short time limit. Ames et al. studied the use of evangelistic and quasi-religious metaphors to encourage volunteer participation in a civic hackathon for disaster relief technologies [5]. In our study of collegiate hackathons, we found similar kinds of excitement generated by social energy, time urgency, an action-oriented "show, don't tell" spirit, and evangelism of technologies by representatives of sponsoring companies. However, the aforementioned corporate and civic hackathons were attended by working adults who already had jobs (and often families), not by undergraduate students.

Unlike civic or government hackathons, collegiate hackathons usually have no explicit mission of encouraging projects to improve civic engagement or social good. And even though collegiate hackathons are free events which are supposed to be nonprofit, they share some similarities with corporate hackathons [7, 14] since funding is often provided by corporate sponsors. Although sponsors do not directly dictate what projects the students work on, they do provide incentives such as free copies of their software or programming-language APIs, volunteer mentors to set up and debug that software, swag such as free company-branded t-shirts and USB flash drives, and prizes for using their software in creative ways in hackathon projects. In addition, some students are attracted to creating projects that appeal to sponsoring companies since they want to obtain internships or full-time jobs there.

To our knowledge, we are the first to study what occurs during an existing student-organized collegiate hackathon, how students perceive these events, and why students choose *not* to participate. In the related StitchFest project [18], researchers created a new specialized hackathon (focused on wearable devices) to broaden participation amongst women in computing, and then studied its 33 participants.

WHAT ARE COLLEGIATE HACKATHONS?

Before presenting our main study, we first provide background information on what collegiate hackathons are, who participates, and how they have grown in recent years. For simplicity, throughout this paper we will use the term *hackathon* to refer exclusively to collegiate hackathons.

A hackathon is a 24 to 36 hour free weekend event hosted in a large indoor open space within a college campus, with tables and chairs for students to set up their laptops (see Figure 2). 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 swag 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 internship or job offers at sponsoring companies [15].

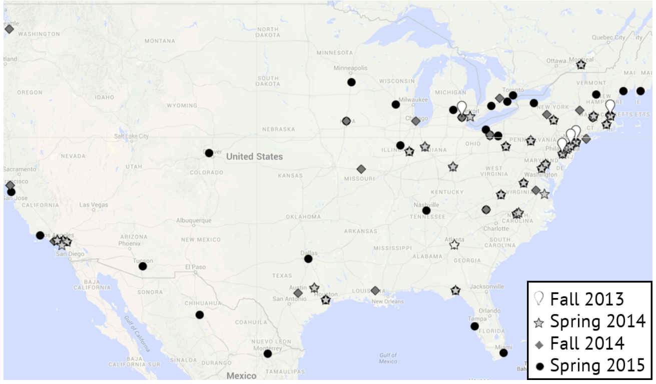


Figure 3. Locations of hackathons sponsored by Major League Hacking during the past four seasons: 5 in Fall 2013, 38 in Spring 2014, 36 in Fall 2014, and 57 in Spring 2015. Hackathons started in the Northeastern U.S. (see white pins for Fall 2013) and spread westward and southward.

Hackathons started in the Northeastern United States in 2010 [10] and have since become pervasive across U.S. college campuses. Figure 3 shows the spread of hackathons sponsored by the official American league, Major League Hacking (MLH) [2], during the most recent four semesters. MLH will host around 150 hackathons in 2015, and 45 out of the 47 top-ranked U.S. computer science departments [1] have hosted their own hackathons. Many more colleges host smaller hackathons that are not registered with MLH. Hackathons are also spreading to other countries, but we do not have detailed data on their growth at non-U.S. campuses.

To give a sense of scale, the five largest and most well-funded U.S. hackathons in the 2014–2015 academic year were TreeHacks at Stanford (with $68,590 in prize money), PennApps at UPenn ($46,060), MHacks at the University of Michigan ($35,670), LAHacks at UCLA ($30,300), and HackMIT at MIT ($22,250). Around 500 to 1,000 students from the U.S. and several other countries traveled to attend these events, which are usually held in indoor sports stadiums. However, most hackathons are not as high-profile as these flagship ones; most attract around 100 students from nearby schools.

The majority of hackathon attendees are undergraduate students. To get a sense of how many hackathons each student attends, we scraped data from Devpost [3], the main website where hackathon participants register their online profiles and

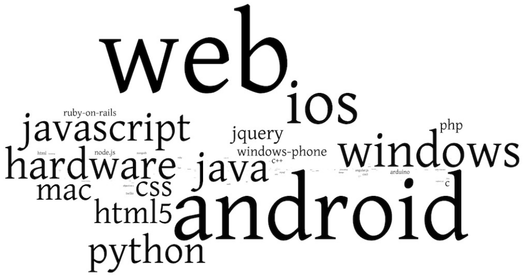


Figure 4. The most frequently-listed interests of 5,669 students who attended the largest 20 hackathons in the Spring 2015 MLH season [2].

submit projects for judging. For the largest 20 hackathons in Spring 2015, there were 5,669 registered students (an average of 283 per hackathon). Of those, 60% were first-time attendees, 7% had attended at least 5 prior hackathons, and the most active 1% had attended at least 10 prior hackathons.

Figure 4 shows the most frequently-listed interests on participants' Devpost [3] profiles. Each student listed an average of 3.4 interests, mostly related to software technologies. As of Spring 2015, Web programming was the most popular, followed by mobile device programming on Android and iOS. Popular programming languages include JavaScript, Python, Java, and PHP. Some participants were also interested in interfacing with hardware such as the Oculus Rift VR headset, usually provided at hackathons by MLH, university labs, or company representatives to advertise their products.

METHODOLOGY

The goal of this study was to characterize students' expectations for and experiences at hackathons, so we gathered data using semi-structured interviews and direct observations at a hackathon. We also surfaced the opposite perspective – why students do *not* participate in hackathons – via a survey of 1,473 college interns at a large U.S. software company.

In-depth study of six hackathon participants

We selected six subjects from the population of undergraduate students at our home institution (a mid-sized Ph.D.-granting university in the United States) who had registered to attend our university's annual hackathon. To maximize diversity of perspectives, we strove for a balance both in gender (3 female, 3 male) and in level of prior hackathon experience (3 with no experience, 2 with some experience, 1 with lots of experience). Here are the pseudonyms for our study subjects, coded by both gender and ascending experience level:

	Male			Female		
Pseudonym:	M1	M2	M3	F1	F2	F3
# hackathons attended:	0	3	9	0	0	2

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

The first author of this paper, an undergraduate student who is a peer of these subjects, conducted three sets of 30-minute interviews with each subject – one week before our university's hackathon, one week afterward, and one month afterward. He also observed them at work during the hackathon.

Pre-hackathon expectations interview: This interview was conducted the week before the hackathon, and its purpose was to assess subjects' expectations leading up to the event. We asked the following questions, although each conversation was semi-structured and veered toward 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 this hackathon?
- Do you already have project ideas or team members?
- Who are you hoping to learn from?
- Are you nervous at all about any aspects of this hackathon?

Direct observations at hackathon: To supplement the interview data, the first author attended our university's hackathon, observed those six students at work throughout the event, and took field notes. This was a 36-hour hackathon that started on Friday evening at 5pm and ended on Sunday at noon (with 7 extra hours for sleeping). There were 84 participants (24% women) from eight universities, and nearly everyone was an undergraduate student. The researcher struck a balance between being unobtrusive and inquisitive, using his judgment of etiquette from attending previous hackathons.

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

- What project did you end up working on?
- Did you change project ideas or groups mid-way?
- Who did you end up learning from?
- What did you end up learning? Anything unexpected?
- What was the most memorable thing about this hackathon?
- How did it 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 retained one month afterward.)
- How you think you will use those skills in the future?
- What criticisms do you have of hackathons after attending? Did you remember any discouraging experiences?

Data coding and analysis: As each round of interview and observation notes arrived, the research team iteratively coded them to identify major themes and to inform the questions to ask in the next round. Our team consisted of an undergraduate student with prior hackathon experience and a professor with no hackathon experience.

Interviews with four student critics of hackathons

To get an opposing perspective on hackathons, we contacted four students and recent alumni we knew who were critical of these events: two men and two women. All have attended hackathons in the past few years. One of the men leads our own university's student hacking group. And the two women were both co-organizers for several of the largest U.S. hackathons. We chose these particular subjects because we wanted to get the opinions of expert insiders rather than outsider critics. We conducted 30-minute interviews to discuss the negative aspects of hackathons. These interviews motivated the design of our survey, which we now describe.

Survey of 1,473 software development interns

After analyzing data from our ten interview subjects, who have all attended hackathons, we wanted to get a broader range of criticisms from students who *chose not to attend* collegiate hackathons. To do so, we sent an online survey to 1,473 software development interns (all college students) working at a major U.S. software company in summer 2015.

Our survey asked the following questions and prompted students to respond only if they were critical of hackathons:

- What is your gender?
- If you have *never attended* a collegiate hackathon, what factors discouraged you from attending?
- If you have attended hackathons but *did not enjoy the experience*, what aspects of the event felt discouraging to you?

We kept the survey short to target the information we needed to supplement our main interviews and observations. We received 256 responses (17% response rate) and performed inductive analysis to code for recurring themes (Tables 1 and 2). 84 respondents identified as female (33%). For privacy reasons, we could not ask students for their college, race, ethnicity, or country of origin, but IP addresses showed that 38 respondents (15%) came from non-U.S. office locations.

FINDINGS

We now present findings from our interviews, direct observations, and survey. We explore why students choose to attend (and not to attend) hackathons, what makes the experience engaging, how and what they learn, and the main criticisms that both attendees and non-attendees have about these events.

Motivations for attending

Our subjects primarily viewed the hackathon as a weekend social event. F2 felt that *"sometimes it's worth it to push off schoolwork [for a weekend] for the joy of going to a hackathon."* Prior to attending, M1 and F2 already formed a team with friends from classes, while the other four subjects expected to find teammates there. In the pre-hackathon interview, no one was nervous about finding teammates or a project to work on. In fact, M1, M3, and F3 mentioned the appeal of meeting new people there. Hackathons are intercollegiate social events where schools organize buses to take students to hackathons at 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:

> *"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."*

Experienced subjects were excited about technical opportunities. For instance, M3, who had attended 9 prior hackathons, mentioned that he was excited to gain insider access to software development kits, APIs, and hardware that the sponsoring companies provide, 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 Arduino electronics boards.

Surprisingly, none of our interview subjects mentioned being excited about the competitive aspects of the hackathon, even though prizes were offered. M2 mentioned the exact opposite: *"Hackathons are a sort of intellectual free-for-all. You go there to learn in the context of the computing/hacking community. Focus is not, or should not be, on prizes, but rather to go to make something cool."* This sentiment differs from the feeling at intercollegiate programming competitions, where the main goal is to win by solving a fixed set of problems.

Environment: intensely concentrated in time and space

A hackathon provides a concentrated and relatively distraction-free venue for students to focus on completing a specific task. Everyone is gathered in one central location for a fixed time period with the same goal: to create a demoable software prototype. Attendees appreciated the excuse to get away from the usual distractions of campus life, and those who traveled from other schools often had nothing else to do but hack. The time pressure of creating a working demo by the end of 36 hours also motivated students to concentrate.

The most salient difference between working at a hackathon and in class (or at a student maker club) 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. Also, rather than being curriculum-oriented, a hackathon is project-oriented. M3 said that hackathons are about *"the pursuit of making, not mastery."* And F2 said:

> *"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 [my project] 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."*

In the spirit of minimally-guided instruction [12], students completely determine what and how they learn in a hackathon; professors are nowhere in sight. F3 mentioned, *"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."*

On the flip side, there is no formal structure or pedagogy. M3 mentioned that, unlike classes, there is almost no time for reflection and expert feedback in a hackathon, since there is no professional instructor whose job it is to give feedback. He continued, *"The main source of feedback is the current functionality of your project, and your peers' perceptions of it."*

How and what subjects learned at the hackathon

Since collegiate hackathons take place within a school context, learning emerged as a major theme in both interviews and observations. The specifics of what the subjects built at the event were not as salient as how and what they learned.

Incidental learning: Subjects learned at the hackathon as a side-effect of trying to get their project to work. During pre-hackathon interviews, only F1 mentioned that her goal was to attend to explicitly learn a new skill – Web programming from Codecademy [X] tutorials – which she ended up doing. Everyone else focused the conversation on what they wanted to *do* at the hackathon rather than what they wanted to learn. A pervasive example of incidental learning was students learning how to use software tools such as the Git version control system, how to their upload 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.

Opportunistic learning: Participants often learned opportunistically – taking advantage of the short-lived opportunities at the hackathon to direct their learning toward sometimes-unexpected paths. For example, M3 started working on a Web programming project, an RSS feed generator, but then saw that Pebble smartwatches were available to borrow at the hackathon venue. He quickly borrowed one and switched his project to hacking on a smartwatch app, opportunistically learning the Pebble API by reading online tutorials. But afterward, he mentioned that one downside of this approach is that he would not get to reinforce those skills later since he did not own a Pebble watch himself.

Peer learning: During pre-hackathon interviews, subjects described their perceptions of how learning occurs at hackathons. M1 said that the *"learning style consists of goal-oriented peer learning,"* and M3 said that *"hackathons specialize in small group learning,"*. Hackathons foster collaborative learning by having students work in teams; many do not allow solo projects to be submitted for judging. Although several mentors were present, sent by companies or student hacking groups, there were not enough available to help everyone. Novices in particular were reluctant to approach mentors for help, so they ended up learning more from peers.

Even students who worked alone at the hackathon benefited from collaborative learning. 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, and even if students were working on class assignments together, it would be harder to get them to take a break to help her with unrelated topics.

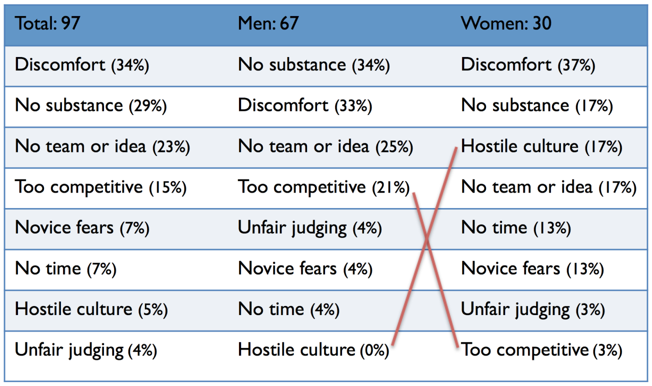
During post-hackathon interviews, several subjects mentioned peer learning as a memorable part of the event. F2 learned how to work better in groups, how to split up work, and how to integrate different components with intense time pressure. She also did a lot of pair programming to see how others approached 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. The lack of authority figures at the event meant that goofing off was socially acceptable and not at odds with learning.

Supplementing in-class learning: M1 went to the hackathon with a friend and planned to use that time to make progress on a class project, which they ended up doing. Since it was his first hackathon, he was apprehensive about not being able to meaningfully contribute to an original project, but he still wanted to attend to experience the event. Working on an existing class project was a good compromise. Some novices used the hackathon as a forcing function to learn basic programming skills like they would in a class rather than creating a project for the sake of competition. F1, along with a group of sophomore optics and physics majors, spent the hackathon working through online Codecademy tutorials on Web programming to learn basic HTML, CSS, JavaScript, and Python, with no intent to submit a project for judging.

Lasting impacts of attending the hackathon

During our final round of interviews (one month after the hackathon) we asked subjects about what knowledge they retained from the event. Our subjects did not vividly recall the specific skills they learned there. Many remembered the event as a blur, with only vague recollections of the general kinds of topics they had learned, such as Web programming. They attributed this lack of retention to the fact that learning was 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 in a class. The main 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 summer internship.

The most salient lasting impact was not about specific technical skills but rather on students' self-confidence about their own coding abilities. Only the three female subjects reported this effect (although our sample size is too small to make any generalizations). 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 building her self-efficacy in the process.



Total: 97	Men: 67	Women: 30
Discomfort (34%)	No substance (34%)	Discomfort (37%)
No substance (29%)	Discomfort (33%)	No substance (17%)
No team or idea (23%)	No team or idea (25%)	Hostile culture (17%)
Too competitive (15%)	Too competitive (21%)	No team or idea (17%)
Novice fears (7%)	Unfair judging (4%)	No time (13%)
No time (7%)	Novice fears (4%)	Novice fears (13%)
Hostile culture (5%)	No time (4%)	Unfair judging (3%)
Unfair judging (4%)	Hostile culture (0%)	Too competitive (3%)

Table 1. Summary of 97 student responses to the survey question: "If you have attended hackathons but did not enjoy the experience, what aspects of the event felt discouraging to you?" Each column adds up to more than 100% since some responses contained more than one theme. Statistically significant differences between genders are shown in red.

Criticisms of hackathons from prior attendees

From interviewing our six subjects, four additional students who were critical of hackathons, and analyzing survey responses, we discovered eight main categories of criticisms. Table 1 shows the frequency of each category that the research team coded from 97 survey responses to the question: "If you have attended hackathons but did not enjoy the experience, what aspects of the event felt discouraging to you?" It also shows which were mentioned more by each gender[1].

Discomfort: 34% of respondents complained about physical discomfort when attending these events, citing 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 unappealing. Also, they experienced stress and discomfort due to the time pressures of working intensely over an entire weekend and the often-loud ambiance of hundreds of students together in a crowded space.

No substance: 29% of respondents mentioned how hackathons incentivized building flashy app demos rather than creating something more substantive and lasting using computer science principles, since judges were more likely impressed by cool-looking demos. One respondent wrote, *"Too much emphasis on mobile/web. I also prefer the technical side as opposed to presentation/visuals."* Another wrote, *"the end result is usually a buggy, fake [mobile or Web app] frontend with fake data, and presentations are only there to hide this fact - If I have an idea that's worth building, it's definitely going to take more than 48 hours to put together."* Also, hackathons do not teach rigorous software engineering practices; there are no instructors to critique the technical architecture of one's project, or how maintainable or robust the code is. One student critic we interviewed said, *"Hackathons encourage unhealthy work ethics like pulling all nighters, 'crushing out' code, sloppily throwing things together just so they work at the last minute. All of these bad ethics are being encouraged at an event that's funded by recruiters for companies, just seems sort of iffy to me."*

No team or idea: Attendees who did not come to hackathons already with a team or idea often had a hard time finding teams, especially if their interests did not align with current tech trends. One wrote, *"I have always found it difficult to find partners for hackathon projects at my school. I center many of my hackathon projects around compilers and my peers don't seem to like that."* Teams sometimes dissolved mid-way through the event due to ideas fizzling out, team members being flaky, or team members not getting along.

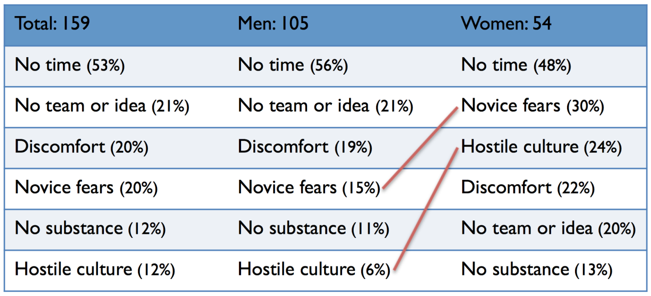
Too competitive and ***Unfair judging***: Although submitting projects for judging is optional, some participants felt like the presence of cash prizes and company recruiters made the atmosphere overly competitive. Also, judges were often peers or company representatives, so some felt that there was a lack of rigor in judging. One respondent summarized these sentiments: *"Many teams go in with full-fledged [project] plans, and people who literally go in to hack and have fun are left behind in an increasingly competitive landscape."* Table 1 shows that men mentioned the "Too competitive" issue seven times more often than women did (21% of men versus 3% of women, chi-square test $\chi^2(1, N = 97) = 3.6, p = .05$). 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. (However, none of our six interview subjects mentioned competition/winning as their primary motivators for attending.)

Novice fears: 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 formal 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. Mentors are practitioners and not trained educators, so they might not be good at explaining fundamental CS101-type concepts. Asking for help can be intimidating, especially as a novice in a room filled with more experienced hackers busy working on their own projects. A novice needs to have a proactive and somewhat-aggressive personality to catch people's attention to get help. Although women were three times more likely to mention this theme than men (13% vs. 4%), the disparity was not statistically significant ($\chi^2(1, N = 97) = 1.3, p = .26$).

No time: Some 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 said, *"Eats up a lot of time. Takes away from potential opportunities to rest and replaces them with a high-stress, exhausting environment."*

Hostile culture: 17% of women cited stereotypical "hacker culture" (well-documented in the literature [X,X,X,X]) as being hostile and discouraging; *none of the male attendees* mentioned culture in their responses. One woman wrote, *"The culture can be a bit unwelcoming and overwhelming."* Several mentioned sexism directly, such as: *"The teams were not willing to take on a less-experienced programmer. Also, I was the only woman out of 100+ male participants, and some*

[1] Although we provided "Other" and "Decline to state" gender options in the survey, all respondents self-identified as female or male.



Total: 159	Men: 105	Women: 54
No time (53%)	No time (56%)	No time (48%)
No team or idea (21%)	No team or idea (21%)	Novice fears (30%)
Discomfort (20%)	Discomfort (19%)	Hostile culture (24%)
Novice fears (20%)	Novice fears (15%)	Discomfort (22%)
No substance (12%)	No substance (11%)	No team or idea (20%)
Hostile culture (12%)	Hostile culture (6%)	No substance (13%)

Table 2. Summary of 159 student responses to the survey question: "If you have never attended a collegiate hackathon, what factors discouraged you from attending?" Each column adds up to more than 100% since some responses contained more than one theme. Statistically significant differences between genders are shown in red.

flirted with me." Another wrote: *"MULTIPLE men came up to me and ASKED ME IF I NEEDED HELP. I have been coding since I was 10, I do not need or want help from creepy men hitting on me."* All three female subjects from our university mentioned culture-related issues during interviews: F1 said that people glorifying pulling all-nighters was unappealing. F2 said that an overemphasis on geeky social games, such as superhero trivia contests, at past tech events made her feel unwelcome. F3 said that she never witnessed any overt discrimination, but she sometimes got strange looks and felt like women participants were perceived as not being as technically capable as the men. Both female student critics (who co-organized large U.S. hackathons) said that since hackathons are student-led, often by 18- to 21-year-old young men, they lack adult supervision to encourage more mature social norms. For instance, project demos have featured immature or misogynistic concepts, and the male-dominated, music-filled atmosphere can feel like a fraternity party.

Why students choose not to attend collegiate hackathons

Since hackathons are growing more prevalent across college campuses (e.g,. Figure 3), we wanted to investigate why students who know about these events choose *not* to attend. Table 2 summarizes 159 survey responses we received to the question: "If you have never attended a collegiate hackathon, what factors discouraged you from attending?" Note that these respondents all knew about hackathons; otherwise they would have not written responses about discouraging factors.

"No time" was the most common factor. Even those who like the idea of hackathons did not want to give up a weekend to attend. One female respondent wrote, *"I love the idea but I am [a] varsity athlete and cannot physically attend the whole time period. It would be interrupted by mandatory practices. And I would not be willing to sacrifice sleep if I had practice the next morning. I would love to attend one however."*

Even though these respondents had never attended hackathons before, nonetheless they still mentioned the same discouraging factors as attendees: no time, no team or idea, physical discomfort, and no substance. The prevalence of these themes did not vary by gender. One possible explanation is that these respondents had heard about what happens at hackathons from peers who have attended, and formed negative preconceptions from their peers' complaints.

Similar to attendee responses, the two kinds of discouraging factors for non-attendees that had strong gender skew were "Novice fears" (30% of women vs. 15% of men, $\chi^2(1, N = 159) = 3.7, p = .05$) and "Hostile culture" (24% of women vs. 6% of men, $\chi^2(1, N = 159) = 9.7, p = .002$). One female respondent mentioned both of these sentiments:

> *"It seemed to be primarily a guy-attended event. Guys I didn't like very much talked loudly about going, while I couldn't find another girl to go with me. Also, I was afraid of having my skills or ideas disparaged. I knew that the guys I mentioned above had more coding experience than I did, and I didn't want to be delegated to watching or make a fool of myself trying."*

Several women mentioned sexism more explicitly, which corroborated similar accounts from prior studies **[X,X,X]**:

> *"I haven't attended a hackathon before, but some of my female friends who have said that they felt looked down upon and marginalized by their male peers. Whenever they were asked questions, they were addressed to the males on the team rather than the team as a whole and the girls were pretty much ignored. This really makes me not want to go to hackathons because I don't want to be treated that way."*

Finally, although "hostile culture" was the least-cited discouraging factor by male students (only 6%), a few did mention it, such as: *"Dislike of not sleeping [pulling all-nighters] being such a big part of CS culture. Discomfort with CS culture in general (see sexism, racism, elitism, etc.) and lack of desire to spend more time immersed in it than necessary."*

LIMITATIONS

We studied students who both attended our university's hackathon and were willing to undergo three interviews. Although we tried to balance for gender (3 female, 3 male) and prior experience (3 first-timers, 2 intermediates, 1 veteran), this sample may still be biased toward more enthusiastic self-selected attendees. To compensate, we obtained opposing perspectives by interviewing four students who were critical of hackathons and surveying 1,473 college interns at a large U.S. software company about their negative perceptions.

The researcher who conducted the interviews and observations was an undergraduate with prior hackathon experience. While this gave the crucial advantage of making subjects feel comfortable talking candidly with a peer, there is the chance that the researcher injected his own biases into notes. To reduce this risk, we instructed him to write down only observations without interpretation, and then coded the data together with a professor with no prior hackathon experience.

We collected data on gender and prior level of experience but did not investigate race, ethnicity, or socioeconomic status. All of our interview subjects were white or Asian, and for privacy reasons we could not ask for other demographic data in our survey of college student interns. Thus, our findings may not generalize to underrepresented minorities in computing, students outside of the U.S., or those who did not attend four-year universities or intern at a major U.S. software company.

DISCUSSION: IMPROVING COLLEGIATE HACKATHONS

Given the recent popularity of collegiate hackathons and the enjoyment that many students derive from them, it is important to take steps to make these events more inclusive and welcoming. Although these have always been desirable goals for any event, they are now becoming urgent as hackathons move toward the mainstream of CS student culture on college campuses. Students now view hackathons not only as social and learning opportunities, but also as opportunities to obtain coveted internship or job offers in lieu of attending traditional career fairs or on-campus job interviews [15]. As companies start recruiting more out of collegiate hackathons, the lack of inclusion at these events means that certain groups (e.g., women, underrepresented minorities) are at a disadvantage when job-hunting. It is no longer a matter of having fun at a niche social gathering; jobs are potentially at stake.

The challenges faced by female students in computer science have been well-documented [17, **?**, **?**]. Our interview subjects and survey respondents corroborated prior findings with their concerns that hackathons embody the most extreme of hyper-geeky environments that implicitly exclude women (and underrepresented minorities). "Hostile culture" was far more often cited by women than men as discouraging (Tables 1 and 2). Subjects F1 and F3 suggested that women-only hackathons [**?**, **?**, **?**] and having more female role models and mentors at existing hackathons would help, although those would only be first steps toward making them feel more welcoming. Another idea is for hackathons to follow the example of successful open-source software communities [**?**, **?**] using clearly-written codes of conduct [**?**] with zero-tolerance policies for harassment or exclusionary behavior.

Making hackathons more welcoming for novices can also broaden participation [18]. In our survey, both men and women cited "novice fears" as a discouraging factor, although women were two to three times more 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. Adding such structure requires high-quality mentors to attend; organizers can recruit from the best of the computer science TA (teaching assistant) and LA (lab assistant) pool at the school. Currently, many 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. For instance, participants noted that some tech talks at hackathons were thinly-veiled advertisements for the sponsoring company's software or APIs [**?**].

To minimize physical discomfort and the unappealing stay-up-all-night nature of hackathons, organizers could mandate that they close down at a fixed time each night. One idea for a 30-hour hackathon could be Friday 6pm until midnight, Saturday 10am until 10pm, and Sunday 10am until 10pm. However, doing so still does not alleviate the "no time" issue, since attendees still need to give up their weekends. Another idea is to spread the hackathon out to a "hack week" [11].

As collegiate hackathons become more popular, they attract more corporate sponsorship. And as sponsoring companies offer more free food and swag, larger cash prizes, and greater prospects of job offers, some critics feel that hackathons are losing their original spirit of making creative technological hacks and are instead turning into full-fledged competitions that glorify winners [**?**, **?**]. Survey respondents cited discouraging factors such as too competitive, unfair judging, and no substance (i.e., making flashy demos to impress judges rather than exploring new technologies). However, our six study subjects did not mention these factors and were overall positive about the event, perhaps due to our university's hackathon being relatively small (84 participants, 24% women) and having almost no corporate sponsorship, so the incentives for competition and winning were minimal.

It can be hard for student organizers to resist the allure of prestigious companies offering increasing amounts of funding to grow these events, but it is important 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 pure maker ethos of small-scale events while growing to hundreds or even thousands of participants? Organizers could strictly limit sponsor involvement, use funding to pay TAs and instructors to come as mentors rather than company representatives, put modest caps on prizes, or follow the example of Duke University's hackathon where prizes were all made as donations to charities [**?**].

Another way to improve hackathons is with more structured follow-up activities. Our study, along with related studies on civic [11] and corporate [14] hackathons, shows that these are exciting one-off events where participants disperse afterward and have a hard time following up to make their projects into something longer-lasting, especially after they get back into the normal flow of school work. Perhaps hackathons could merge with hackerspaces and on-campus maker clubs to enable students to continue working on their projects in a sustainable way throughout the rest of the academic term.

Finally, what can the HCI community learn from collegiate hackathons? The power of these events lies in their ability to generate excitement around shared context and time urgency in a setting away from the usual routine of schoolwork. If organized properly to maximize inclusion and follow-up, these can be great opportunities for teaching HCI principles such as rapid prototyping and collaborative design. Practitioners and researchers are already using hackathons to motivate concentrated bursts of productivity, such as Google Ventures' week-long design consultation sprints **[X]** and the CrowdCamp hackathon at the HCOMP conference for seeding publishable research ideas on crowdsourcing, human computation, and social computing **[X]**. Thus, one direction for future work is to leverage the energy generated by hackathons to teach HCI and design thinking in more engaging ways.

ACKNOWLEDGMENTS

(anonymized for submission)

REFERENCES

1. 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: 2015-09-07.

2. 2015. About Major League Hacking. https://mlh.io/about. (2015). Accessed: 2015-09-07.

3. 2015. Devpost - Hackathons, App Contests, and Hackers. http://devpost.com/. (2015). Accessed: 2015-09-07.

4. 2015. Google Trends. https://www.google.com/trends/. (2015). Accessed: 2015-09-07.

5. 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.

6. Jeffrey Bardzell, Shaowen Bardzell, and Austin Toombs. 2014. "Now That's Definitely a Proper Hack": Self-made Tools in Hackerspaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. 473–476.

7. Alexandra Chang. 2012. Deep inside a Facebook hackathon, where the future of social media begins. *Wired* (July 2012).

8. Carl DiSalvo, Melissa Gregg, and Thomas Lodato. 2014. Building Belonging. *interactions* 21, 4 (July 2014), 58–61. DOI:http://dx.doi.org/10.1145/2628685

9. Sarah Fox, Rachel Rose Ulgado, and Daniela Rosner. 2015. Hacking Culture, Not Devices: Access and Recognition in Feminist Hackerspaces. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. 13.

10. Jon Gottfried. 2014. A brief history of hackathons (lecture video). http://youtu.be/Zr6VPAe9CKU. (March 2014). Accessed: 2015-09-07.

11. Lilly Irani. 2015. Hackathons and the Making of Entrepreneurial Citizenship. *Science, Technology & Human Values* (2015).

12. Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist* 41, 2 (2006), 75–86.

13. Beth Kolko, Alexis Hope, Brook Sattler, Kate MacCorkle, and Behzod Sirjani. 2012. Hackademia: Building Functional Rather Than Accredited Engineers. In *Proceedings of the 12th Participatory Design Conference: Research Papers - Volume 1 (PDC '12)*. 10.

14. Marko Komssi, Danielle Pichlis, Mikko Raatikainen, Klas Kindström, and Janne Järvinen. 2014. What are hackathons for? *IEEE Software* PP, 99 (May 2014).

15. Steven Leckart. 2015. The Hackathon Fast Track, From Campus to Silicon Valley. *New York Times* (April 2015).

16. Silvia Lindtner, Garnet D. Hertz, and Paul Dourish. 2014. Emerging Sites of HCI Innovation: Hackerspaces, Hardware Startups & Incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. 439–448.

17. Jane Margolis and Allan Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. MIT Press.

18. 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.

19. Andrew Richard Schrock. 2014. "Education in Disguise": Culture of a Hacker and Maker Space. *InterActions: UCLA Journal of Education and Information Studies* 10, 1 (2014).

20. Yuling Sun, Silvia Lindtner, Xianghua Ding, Tun Lu, and Ning Gu. 2015. Reliving the Past & Making a Harmonious Society Today: A Study of Elderly Electronic Hackers in China. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. 44–55.

21. Joshua G. Tanenbaum, Amanda M. Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing Technology: Pleasure, Utility and Expressiveness in DIY and Maker Practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. 2603–2612.