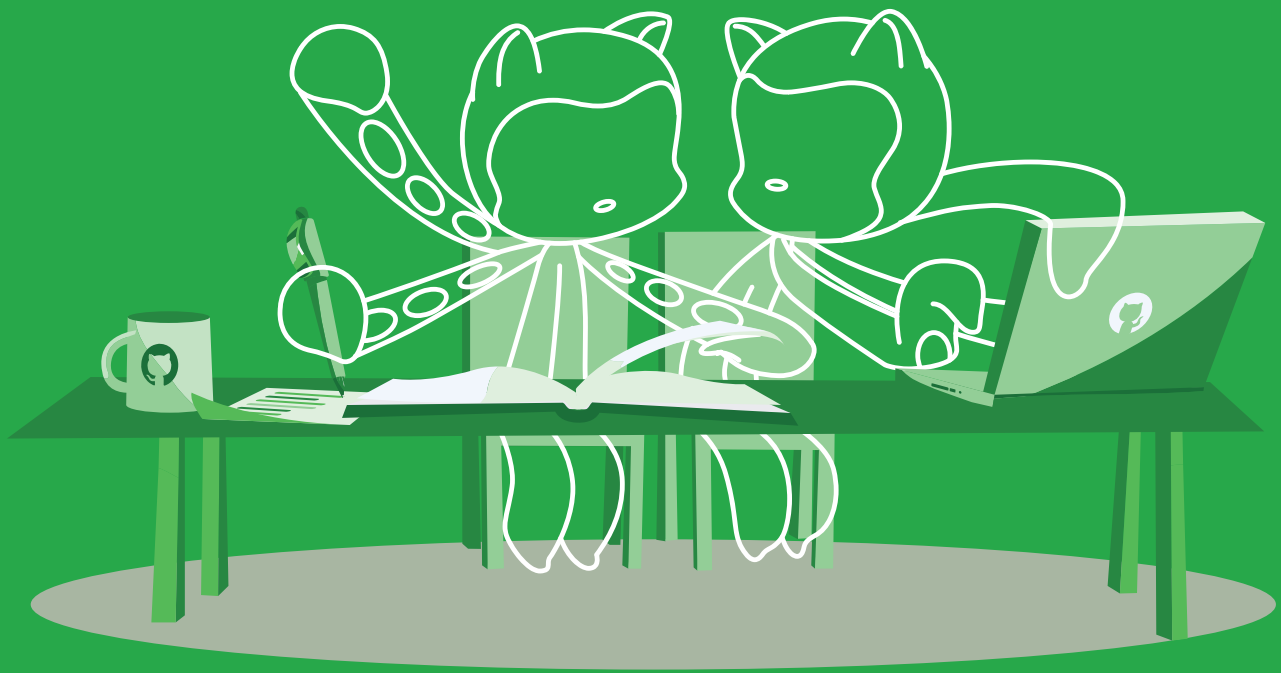
Education

TEACHER STORIES:

How to use GitHub in the classroom



GitHub

Contents

GITHUB EDUCATION

How CS50 uses GitHub to teach computer science

Professor David J. Malan, Gordon McKay Professor of the Practice of Computer Science at Harvard University, is dedicated to offering his students a robust learning experience. This post outlines how he uses GitHub and his own custom tools to build hands-on assignments for CS50 students.

GITHUB EDUCATION

With over 700 students, 80 staffers, and 2,200 participants in their end-of-term CS50 Fairs, CS50 has a reputation for rigor, real-life applications, and engaging material.

At the same time, said Professor David J. Malan, about half of CS50's students typically treat the course as "terminal"—as their one and only course in computer science. So the projects touch on applications in a variety of fields, from social sciences and humanities to medicine and finance.

Malan said of the learning goals of CS50: "We want to provide students with a solid foundation in computer science so that they are well prepared for any field. And also bring to bear some practical skills to that world. So that is actually tied in with our adoption of GitHub this past year."

A gentle onboarding to Git and GitHub

The mental model for cloning, branching, opening pull requests, or pushing can be tricky for newbies just starting out. As a way to onboard students, Malan wrote a command-line tool that wraps a sequence of Git commands called submit50 (https://github.com/cs50/submit50).

They developed submit50 to not “reinvent the wheel” with a new submission system, but to create space for students to grow into comprehensive GitHub use as their learning evolves beyond CS50. Said Malan:

> One goal was to have students graduate, so to speak, with a GitHub account. And even though they don’t keep their work in public portfolios for the course, the hope is that they’ll have at least enough understanding of GitHub that they can use it for personal projects after the term ends.

CS50 course outline

Student workflow for submit50

CS50 uses the structure of one branch per problem, and students engage Git and GitHub from the command line.

First, they run a command in their directory on a Linux system with a folder they wish to submit to CS50's servers. The student then runs submit50 foo where foo is the unique identifier for that assignment.

submit50 models how Git and GitHub work while abstracting away some of the complexity. Malan explained:

> Behind the scenes we show them the command so that through a bit of osmosis, they can infer what's actually going on.
>
> We clone their repo, which lives in our submit50 organization. So we have full administrative rights, and students have push and pull privileges only.
>
> The submit50 script clones that repo into a temporary directory. We do the equivalent of rm -rf to blow away whatever is currently in there, and then git-add the entire contents of their current working directory into that repo, and then tag it with a tag that corresponds to the problem's name, and then push it to the server.

Project-based assignments, real-life applications

An example assignment is "C$50 Finance," where students build an application to simulate stock trades using Python with Flask, HTML, CSS, and SQL.

Students create tables with user accounts, who can then buy and sell stocks. The application queries Yahoo! Finance for stock quotes, almost in real time.

Malan is delighted to see the different personal touches students add to their projects, both functional and aesthetic:

> It's a fun opportunity to introduce students to the aesthetics of web design. Invariably, the first thing students do is customize the aesthetics of the site, and then certainly there are differences in features, and we're fine with that. The assignment requires ultimately that they add a personal touch, so any feature that's not been enumerated by us, they're welcome to do so, [so] long as it's of reasonable scope. So we'll get different outputs from that as well.

Rituals of success

All students exhibit their final projects at an end-of-semester "CS50 Fair." About 2,200 people attend to see the student demos.

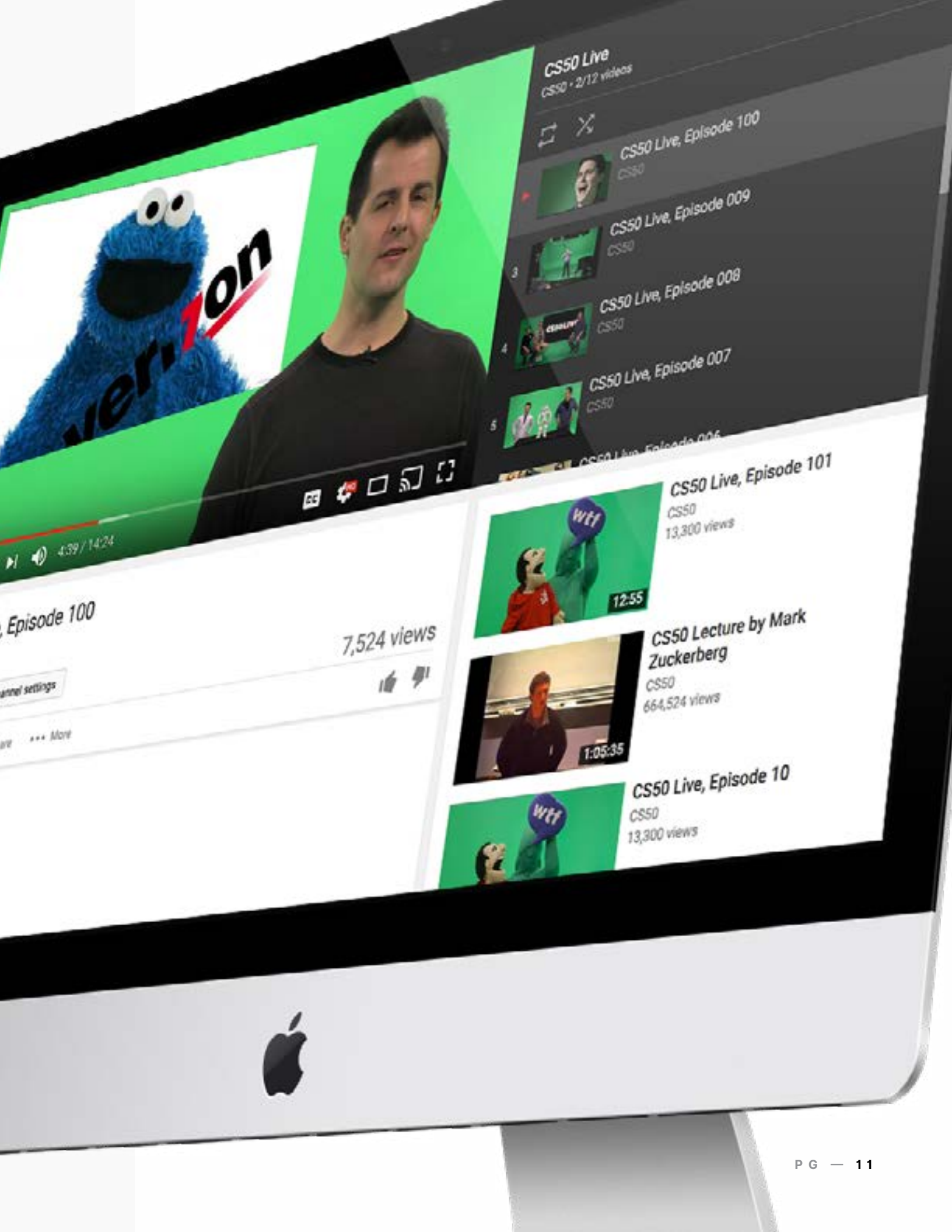
All students exhibit their final projects at an end-of-semester "CS50 Fair." About 2,200 people attend to see the student demos.

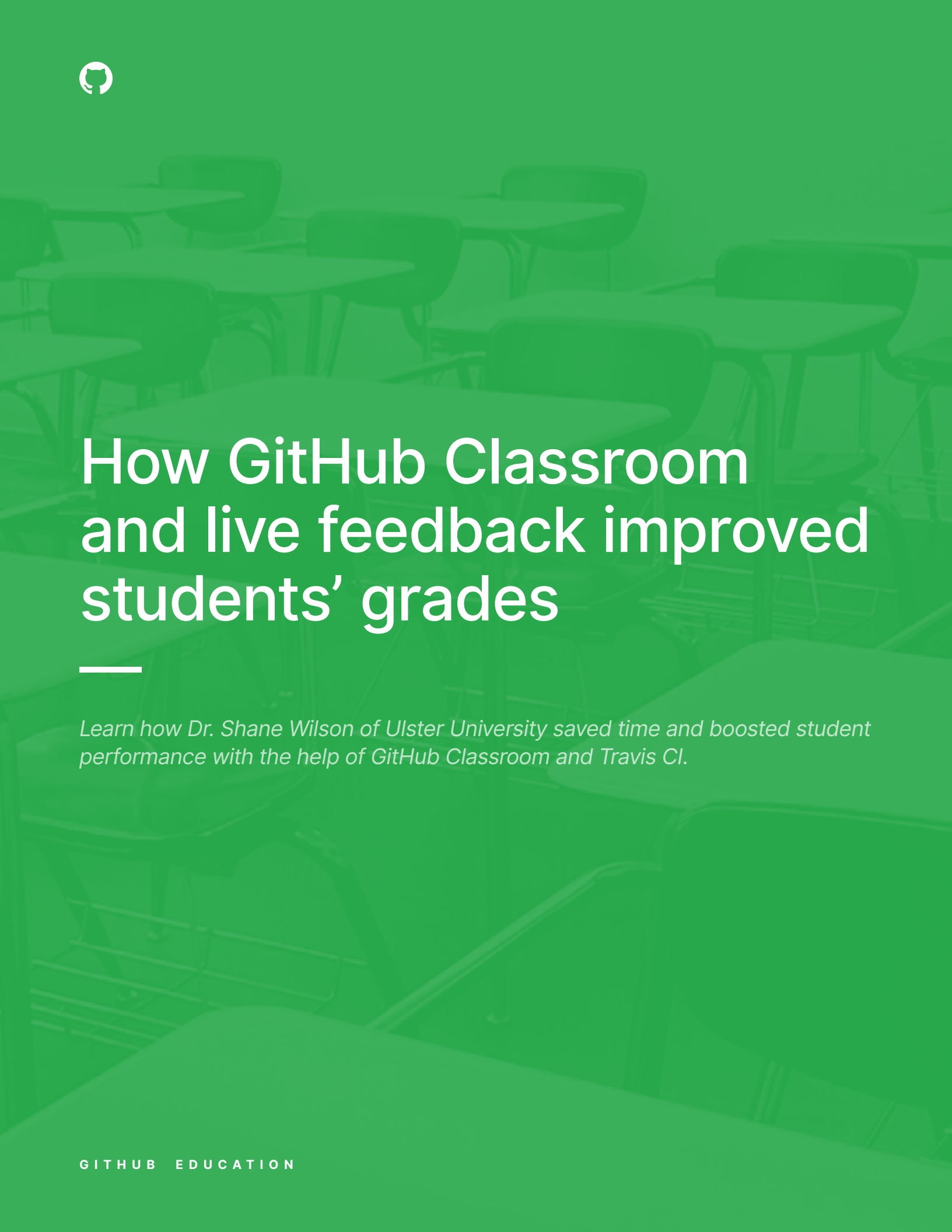
> It's a fun way to delight in how much you've finished, particularly if [you were] new to programming just months prior. And it perhaps creates a bit of social pressure too. You know you're going to be showing this not just to your Teaching Fellow, but to your classmates, and you want to be proud of it. And so, hopefully, that incentivizes all the more of a strong finish.

Computer science and tech news = CS50 Live

Pushing beyond the boundaries of the traditional classroom, Malan connects the course materials with the news in a kind of “Daily Show” for technology, called “CS50 Live.”

Malan and the crew of Teaching Fellows take up current events, like Apple’s buggy implementation of OpenSSL, and dig into the code on the show.


CS50 Live
CS50 • 2/12 videos
CS50 Live, Episode 100
CS50
CS50 Live, Episode 009
CS50
CS50 Live, Episode 008
CS50
CS50 Live, Episode 007
CS50
4:39 / 14:24
, Episode 100
7,524 views
annel settings
••• More
CS50 Live, Episode 101
CS50
13,300 views
12:55
CS50 Lecture by Mark Zuckerberg
CS50
664,524 views
1:05:35
CS50 Live, Episode 10
CS50
13,300 views



How GitHub Classroom and live feedback improved students' grades

Learn how Dr. Shane Wilson of Ulster University saved time and boosted student performance with the help of GitHub Classroom and Travis CI.

GITHUB EDUCATION

With some pre-semester planning, Dr. Shane Wilson was able to decrease the time he spent grading by 80 percent and increase the passing rate of his course from 76 to 90 percent. How did he help students get better grades and learn critical skills, reduce grading time, and scale up to hundreds of students? He combined GitHub Classroom with Travis CI, a service that can automatically generate builds and run tests.

Wilson teaches a 12-week module in object-oriented programming with C++. His students are typically second year students preparing for internships they'll have in their third year. "They are usually attending interviews over the 12 weeks I teach them," Wilson said. To support them at this critical point, he helps them gain confidence in modern development workflows. "It's a real bonus for them to talk about using these tools in their student projects."

An efficient workflow

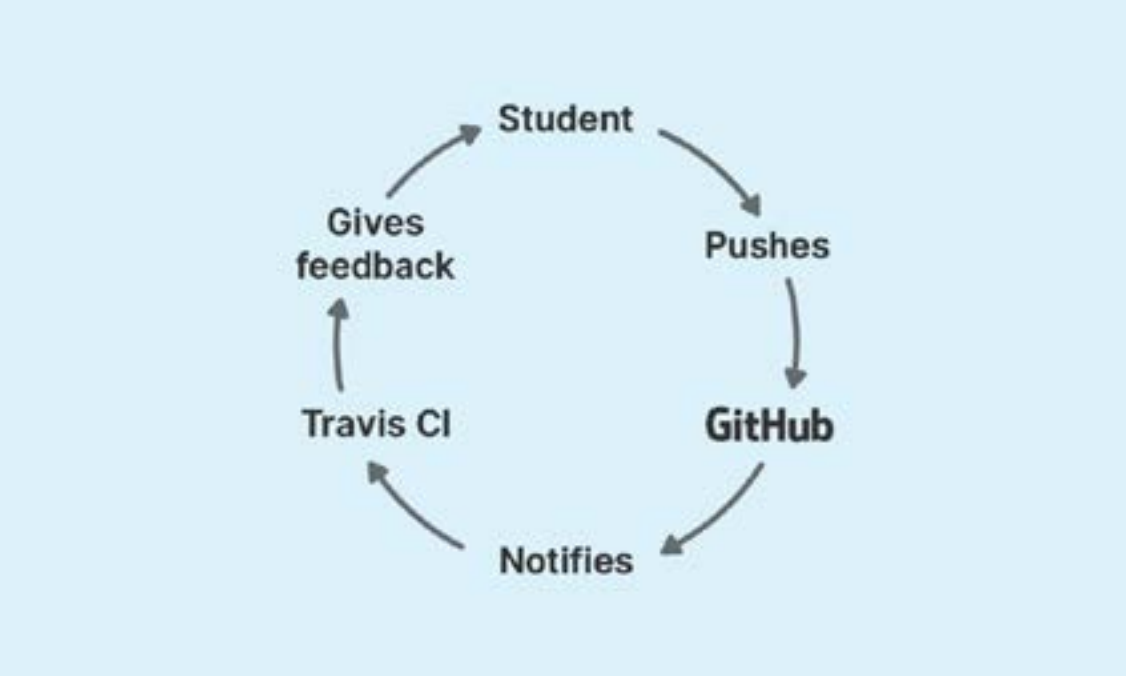
At the beginning of the 12-week modules, Wilson introduces his students to GitHub and Travis CI. In GitHub Classroom, he sets up an assignment with a template repository that contains a README with instructions for students, skeleton C++ classes, and a test suite that will run against the students' code. The test suite is the

core of an assignment repository: The students' grades are set, in part, by what percentage of the tests pass when the assignment is due.

Along with starter code and tests, the repository also has configuration files, including a .travis.yml file. A Travis CI configuration file looks a bit like this:

```
language: cpp
compiler:
 - gcc
script:
 - make test
```

This configuration file in the repository tells Travis CI how to set up and run the tests. Since teachers receive additional organization features as a benefit of joining GitHub Education—like creating individual private repositories for each student—Wilson connects Travis CI to his module's GitHub organization. So when a student pushes to their assignment repository or opens a pull request, GitHub automatically notifies Travis CI of the change. In turn, Travis CI clones the repository and runs the configured commands. For students, the flow of events looks like this:



A diagram displaying the Travis CI workflow. Students push to GitHub, which notifies Travis CI, and then gives feedback to the student, forming a loop.

Wilson said when students get used to using GitHub and Travis CI, they come to appreciate the rapid feedback loop. Students don't have to wait for work to be graded: Passing and failing tests help them gain an understanding of their work each time they push to the repository. Wilson suggested that failing tests can provide "a hint or guidance on what might be wrong," so students can iteratively improve on their work.

Better student results

Wilson explained that this setup has a huge impact on outcomes for students and how he teaches.

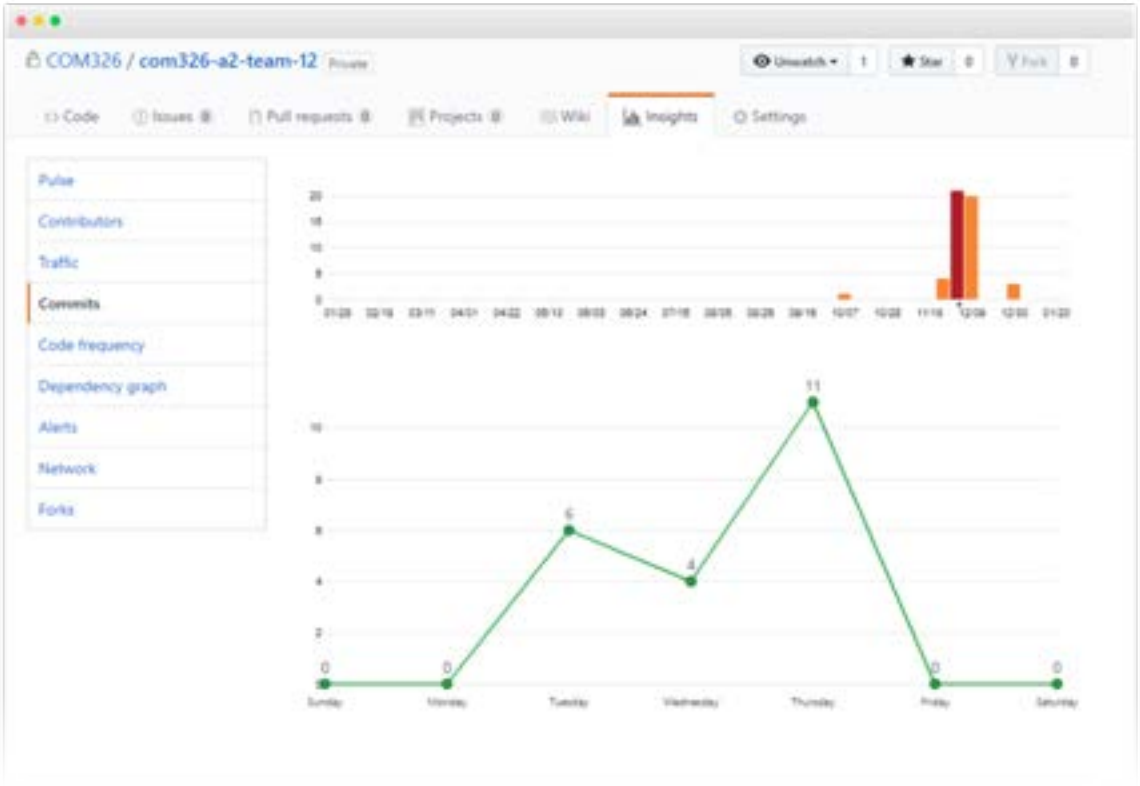
The most important result of this setup is that students in his module are more successful. In the past, only 51 percent of students passed the coursework after the first assignment.

When GitHub Classroom and Travis CI were added to the mix, the passing rate rose to over 87 percent. He reported that the overall progression rate for the module went up from 76 to 90 percent. And students aren't only performing better, they're also more satisfied with the instruction they're getting. Following the adoption of GitHub and other initiatives, Wilson explained, "student feedback on the module was the best I have ever received in 18 years of teaching."

Faster grading with more insights

Wilson has also benefited because GitHub Classroom and Travis CI are saving him time and giving him new ways to guide students. Previously, reviewing and grading students' work would take four to six weeks, but with automated tests, he's spending a few days or a week at most to evaluate students' programming assignments. The statistics that GitHub provides show him when and how students are contributing, presenting new opportunities to help them.

For example, he found that students who were struggling in the course were often waiting to push changes until shortly before the assignment deadline. The "Insights" tab on a repository told him when and how frequently students were adding commits. When students delayed, he could catch that early and give those students a nudge.

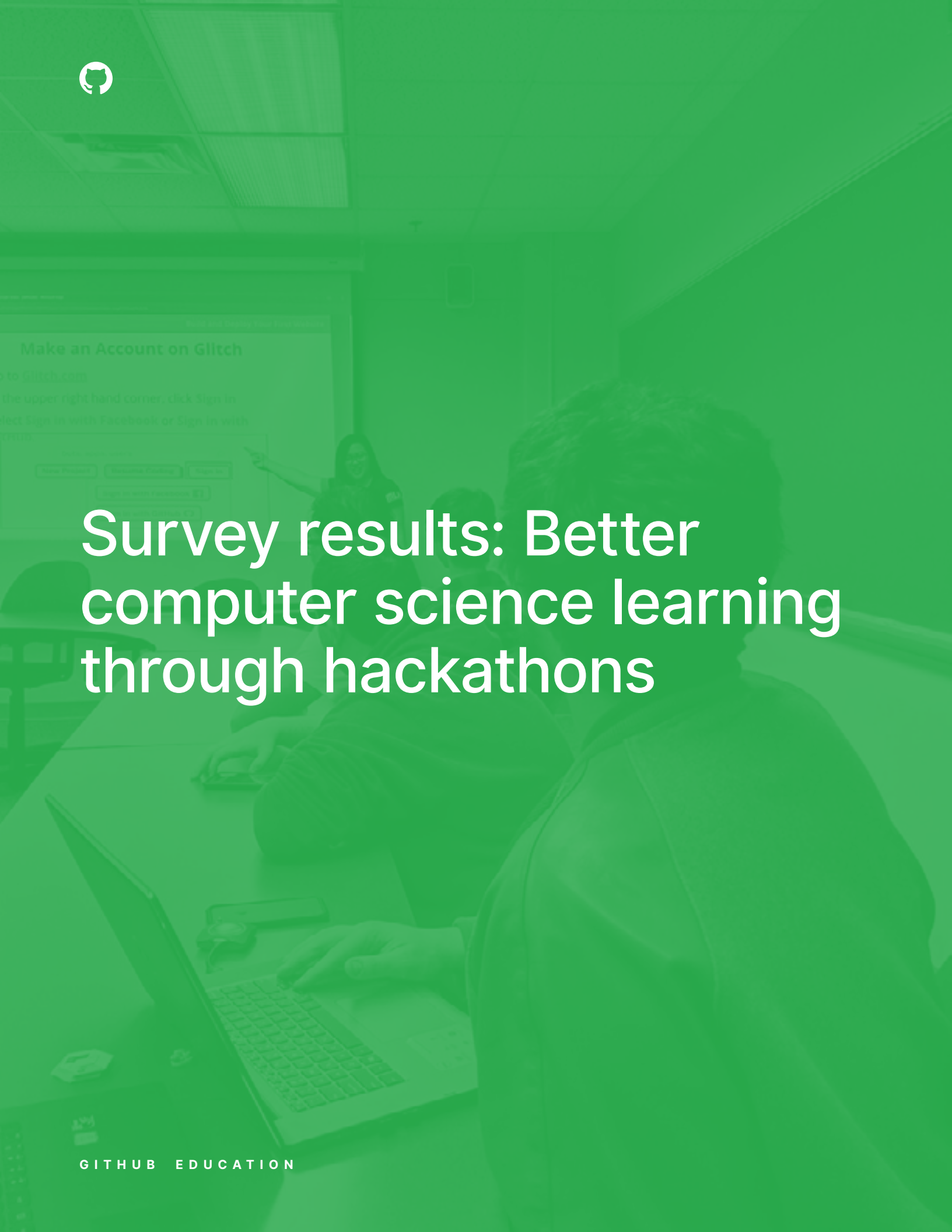


The Insights tab on a repository told Wilson when and how frequently students were adding commits.

Set up for success

Wilson noted that there's still a lot of thought and effort in teaching students this way. He said that students still have to learn Git, which can be challenging to teach—though tools in the GitHub Student Developer Pack, like GitKraken, help. Travis CI can't catch every bug (students can interrupt Travis CI with an infinite loop, for example), so he monitors for stalled builds. Wilson said there's also more thought and work at the beginning of the module, since that's when he writes a test suite, that helps him distinguish each student's performance.

But it's worth the effort, Wilson concluded, "The solution scales easily from a class of five to a class of 500."



Survey results: Better computer science learning through hackathons

GITHUB EDUCATION

The most important thing you can teach your students is how to learn more on their own. Your students start the semester at varying technical levels, and computer science concepts can be difficult to grasp. You won't be able to fit everything you'd like to teach into the short time you have with them. With this in mind, you need a way to boost learning and take your classroom from studying syntax to solving real problems with code. That's where hackathons can help.

Send students to the nearest hackathon

At hackathons, students practice classroom concepts, learn beyond the curriculum, and pick up skills to improve their performance in class. Hackathons take on a variety of formats, but generally, the beginner-friendly events bring developers together

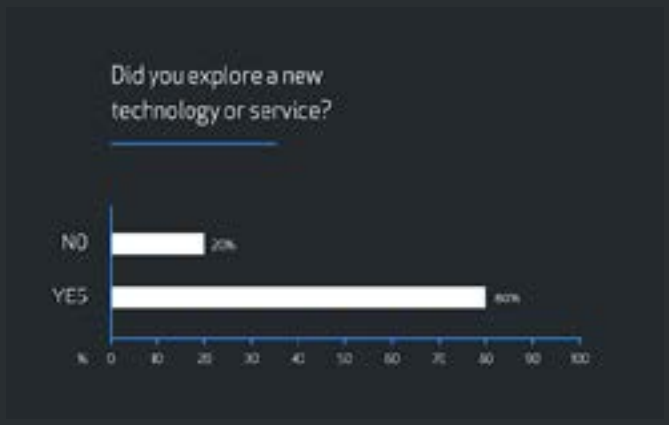
to solve a problem, build a new tool, or get started with a business idea. This typically happens over the course of a weekend. Walk into a hackathon and you'll see groups rushing to finish as much as they can to make their project ready to present before time is up. The event ends with teams presenting their work to a panel of judges, stakeholders, and other participants.

Empower students to master new tools and skills on their own

Hackathons are fun and exhilarating, and survey results from Major League Hacking (MLH) show that attending one can lead to improved participation and performance in the classroom.

Hackathons give students new projects where they can practice the developer tools they use in class. Students improve their mastery when they apply what they've been taught in new contexts outside of school.

Students will also begin to master new tools: 80 percent of students reported exploring a new technology or service while working on their projects.



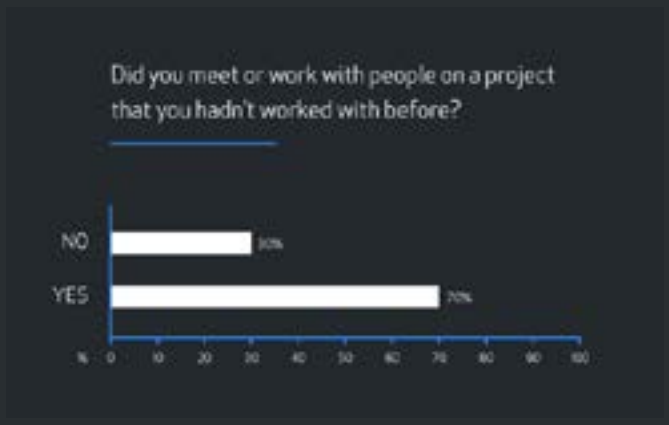
Gaining new experience from a hackathon will improve your students' productivity and spark ideas that will enhance your classroom.

New skills will improve your students' problem solving and decision making. Getting your students to hackathons helps them contribute better solutions, make more realistic projects, and help teach new concepts to their classmates.

These events are an opportunity to learn in an informal setting, and they motivate students to continue learning on their own. When students can adapt to learn outside the classroom, they're better prepared for their future.

Peer-to-peer learning

Each team member in a hackathon brings a different academic and project background, and students exchange a vast amount of knowledge and experience. Seventy percent of students said they met or worked with people they hadn't worked with before, and nearly 90 percent learned something new from a peer.

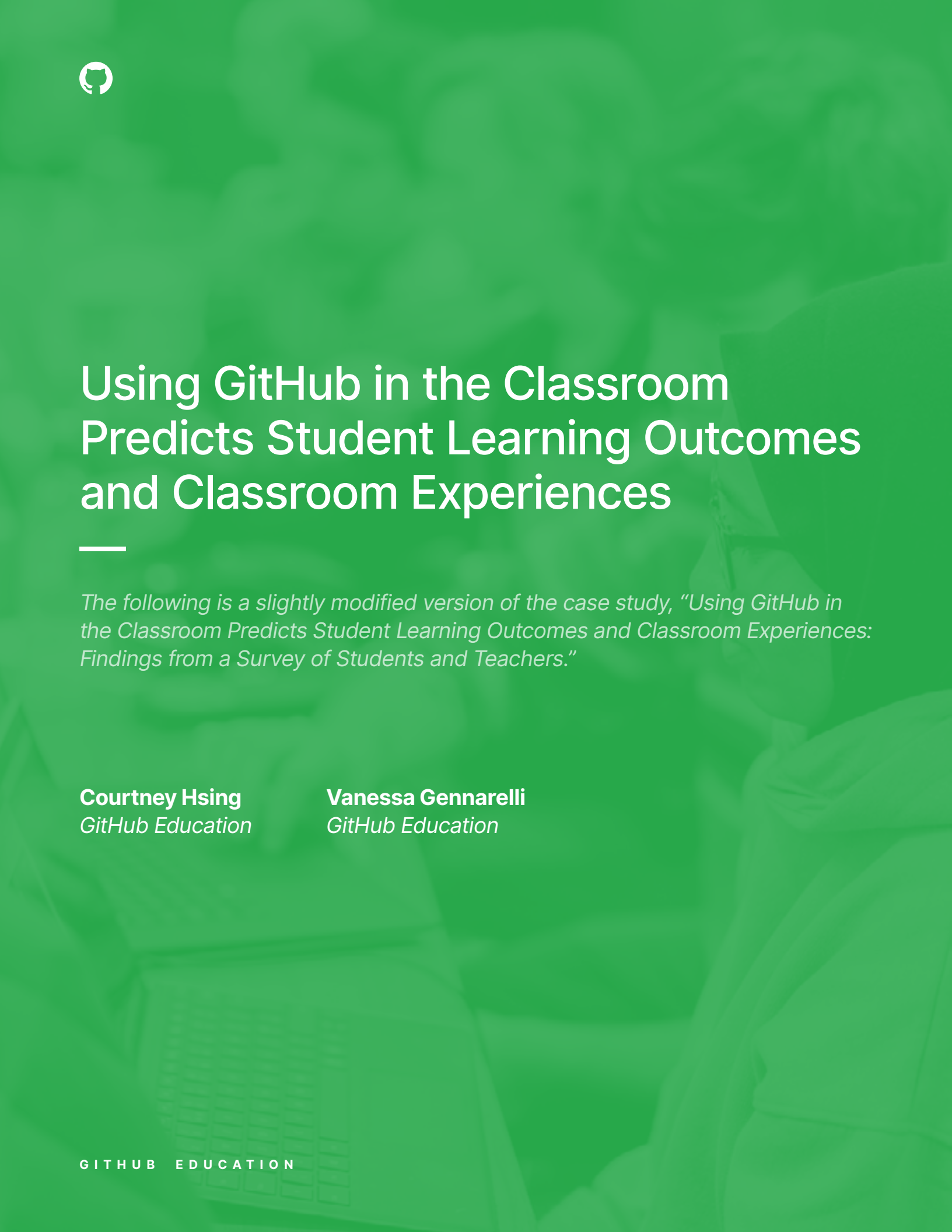


Over the course of a hackathon, participants share tools and transfer skills while working together in teams. Through hackathons, students can become more comfortable articulating technical challenges and sharing knowledge.

Find a hackathon near you

Students across Europe and North America host hackathons through MLH, and there may be a hackathon coming soon near you. Encourage your students to visit MLH (https://mlh.io) to find their next event.


Barq's



Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences

The following is a slightly modified version of the case study, "Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences: Findings from a Survey of Students and Teachers."

Courtney Hsing
GitHub Education

Vanessa Gennarelli
GitHub Education

GITHUB EDUCATION

GitHub is a widely-used software development platform that supports version control, collaborative development, and project hosting. With more than 1.8 million businesses and organizations using GitHub [5], it is recognized as an industry standard. In fact, many developers feel that knowing how to use GitHub is an essential part of securing a job [15]. Currently, an estimated 18,000 educators use GitHub in programming classrooms [14]. Depending on how GitHub is implemented in the classroom, students may rely on GitHub for activities such as submitting assignments, collaborating on group projects, and receiving feedback. Unsurprisingly, the general idea of using GitHub in the classroom has generated interest not only among educators, but also among researchers interested in the implications GitHub has for student learning outcomes. To understand how GitHub can impact the student experience, we will review the general trends uncovered by previous research.

Real-world tools and preparation for the workplace

First, using GitHub in the classroom gives students the chance to learn how to use a popular industry tool. Research supports that one of the main motivations instructors have for implementing GitHub in the classroom is to help students develop familiarity with an important industry tool, thus making students more competitive on the job market [3]. Further, students mentioned the importance of publicly presenting their work on GitHub and that it is common for employers to refer to GitHub during the hiring process [3]. Research also supports that industry collaborations may encourage students to take assignments more seriously, invest greater effort, find assignments more meaningful, and be more motivated to perform well on assignments [9].

Developing soft skills

Next, by encouraging students to use GitHub for collaborative activities, such as peer reviews and group projects, instructors may facilitate students' development of communication, teamwork, and critical analysis skills [2]. Soft skills are important when working in the industry, and computer science and software engineering education have begun to adopt a pedagogy that focuses on soft skills in addition to technical skills [3]. In a study where instructors were interviewed, they noted that using GitHub in their courses effectively introduced students to collaborative practices essential to software development (e.g., cross-team collaboration, peer review) [3]. Further, a survey study found that students preferred the collaborative platform over more traditional individual assignments [8]. One of the key advantages students observed was that GitHub facilitated teamwork and collaboration, allowing students to connect with peers for help outside of the classroom and learn from each other's ideas [8].

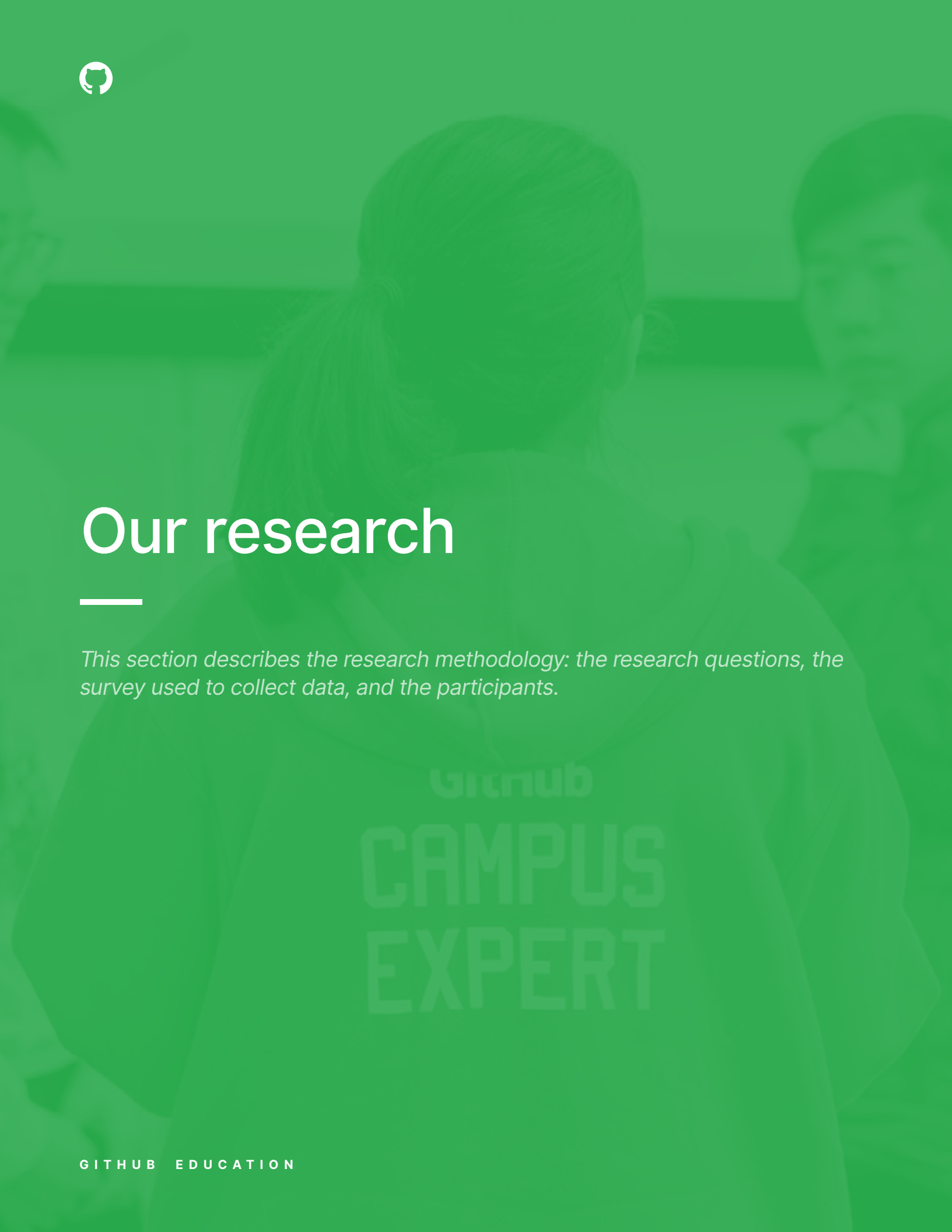
Boosting student engagement

In addition, research supports that using GitHub in the classroom boosts student engagement, or the overall energy that a student invests in the academic experience [6]. By facilitating student contributions and cross-team collaboration, the use of GitHub enables a participatory culture where students feel their contributions are impactful [3]. Further, in one study, researchers interviewed instructors and found that some instructors use students' contributions on GitHub as materials for discussion in class, thus participation is incentivized because student contributions can direct class discussions [15]. Indeed, past research found that one main motivation instructors have for implementing GitHub in the classroom is to offer students the opportunity to directly influence the course and contribute to course materials (e.g., via pull requests) [3], creating novel ways for students to be involved in the classroom.

Increased sense of belonging

Lastly, although past research has not directly addressed this specific idea, using GitHub in the classroom may strengthen the sense of belonging students feel within the classroom because it allows students to collaborate and contribute to each other's learning process. Research finds that more collaborative projects where students work closely with peers can foster a stronger sense of belonging [7]. Importantly, students' sense of belonging can positively predict variables tied to academic success, such as intrinsic motivation and academic self-efficacy [4]. Self-efficacy, the degree to which individuals feel capable of accomplishing a task, is essential to student success because it can protect students' confidence and help them persist when they encounter setbacks [11]. In addition, GitHub supports interactions with individuals in the industry [3] and may promote a sense of belonging to the field. For instance, in a past study, an instructor reported that unregistered students and other GitHub users visited and "starred" their class's public repositories, establishing a connection between students and others in the field [15]. Research suggests that improving students' sense of belonging can have a protective effect on retention and increase achievement motivation [13, 10].

Building on previous research, we investigated how students in classrooms that used GitHub, as opposed to classrooms that did not use GitHub, differed across key variables. Our research deviated from previous studies in several ways. Notably, to date, studies on the use of GitHub in the classroom have relied on data from qualitative studies that mostly focus on instructors' perceptions of student learning. Although these approaches have highlighted important issues, trends, and ideas, we aim to build a deeper understanding of students' experiences and also draw comparisons between the students' and instructors' perceptions. Lastly, we hope to extend our understanding of how GitHub shapes student learning outcomes and experiences by measuring additional relevant outcome variables.



Our research

This section describes the research methodology: the research questions, the survey used to collect data, and the participants.

Research questions

Question one: Does using (versus not using) GitHub in the classroom predict students' learning outcomes?

Past research suggests that using GitHub in the classroom may boost student engagement [6], and encourage class participation [15]. In the survey, we asked students to rate how much they learned about teamwork and collaboration, popular tools, and project management from 1 (very little) to 5 (very much). We also asked instructors to rate their students using the same survey items and scale. We hypothesized that, based on student and teacher responses, using (versus not using) GitHub in the classroom predicts learning more about collaboration, popular tools, and project management.

Question two: Does using (versus not using) GitHub in the classroom predict students' feelings of preparation for the future?

Research suggests that collaborative activities, such as peer reviews and group projects facilitated by GitHub, may strengthen students' development of skills important when working in industry in the future [2]. In the survey, students rated how much 1 (very little) to 5 (very much) their course has prepared them for a future internship or career, being a part of the developer community, developing a portfolio of their work, and taking more advanced courses. We hypothesized that, based on student and teacher responses, using (versus not using) GitHub in the classroom predicts feeling more prepared for all measured domains.

Question three: How is the implementation of GitHub in the classroom associated with students' classroom experiences?

Using GitHub in the classroom may strengthen the sense of belonging students feel within the classroom. Research finds that more collaborative projects where students work closely with peers can foster a stronger sense of belonging [7]. In addition, GitHub supports interactions with individuals in the broader developer community and may promote a students' sense of belonging to the field. We predicted that, using (versus not using) GitHub in the classroom predicts a greater sense of belonging both in the classroom and in the field.

To assess students' sense of belonging in the classroom and in the field, students were asked to rate the following items using a scale of 1 (strongly disagree) to 7 (strongly agree), and we created a composite score by averaging ratings across each set of three items. Items were adapted from a previous study [12].

Belonging in the classroom:

> 1. My contributions in my course are valued.
> 2. I feel comfortable in my course.
> 3. People in my course accept me.

Belonging in the field:

> 1. I see myself as part of the developer community.
> 2. I feel that I am a member of the developer community.
> 3. I feel a sense of belonging to the developer community.

Question four: How does the design of implementation predict outcomes related to classroom learning?

Lastly, because GitHub is a tool with diverse features and can be implemented flexibly in the classroom, we wanted to examine

whether the design of implementation can predict outcomes related to classroom learning. First, we assessed how the number of features students used in the classroom is associated with positive learning outcomes and feelings of preparation for the future. We expected that using more GitHub features in the classroom is associated with greater positive learning outcomes and feelings of preparation for the future. In addition, we investigated how using GitHub as a tool to deliver feedback may predict the perceived effectiveness of feedback.

GitHub allows students to receive line-specific feedback from their instructors or peers via pull requests. Because detailed informational feedback tends to increase student self-efficacy and learning strategies [1], relying on GitHub may make students better learners who use feedback effectively. We measured students' perceptions of instructor and peer feedback by asking them to rate the following items from 1 (strongly disagree) to 7 (strongly agree)

Instructor feedback:

> 1. I understand the instructor's feedback.
> 2. I pay attention to the instructor's feedback.
> 3. I use the instructor's feedback effectively.
> 4. I find the instructor's feedback helpful.
> 5. The instructor understands my needs as a student.

Peer feedback:

> 1. I understand feedback from peers.
> 2. I pay attention to feedback from peers.
> 3. I use feedback from peers effectively.
> 4. I find feedback from peers helpful.

We hypothesized that, based on student and teacher responses, using (versus not using) GitHub to deliver feedback generally predicts greater effectiveness of peer and instructor feedback. However, we did not have specific predictions for how using (versus not using) GitHub to deliver feedback predicts the extent to which students pay attention to instructor and peer feedback, because it is unclear how the manner in which feedback is delivered would shape students' motivation to attend to feedback.

Methodology

Two surveys were designed to collect student and instructor responses. The student survey assessed students' perceptions and experiences in the classroom, and the instructor survey asked instructors to report on their impressions of students' learning experiences.

We invited 459,558 students (9.9 percent high school; 69.4 percent college; 3.3 percent non-traditional; 17.4 percent other) and 17,806 instructors (15.3 percent high school; 62.6 percent college; 2.0 percent non-traditional; 20.1 percent other) to participate in our study. Recipients had verified student or instructor status through GitHub and received a link to the online survey via email. In addition, to be eligible for the survey, the recipient must have either taken or taught a class with a programming component in the past semester. 7,530 students

and 300 instructors who either used or did not use GitHub in the classroom during the most recent semester completed our survey. The response rate of student recipients was 1.6 percent and the completion rate was 82.1 percent. The response rate of instructor recipients was 1.7 percent and the completion rate was 85.0 percent. Tables 1 and 2 provide more information on the student and instructor participants' demographic information.

	Students	Instructors
Used GitHub in the classroom	37.2%	67.0%
Did not use GitHub in the classroom	62.8%	33.0%

Table 1: Student and instructor respondents' use of GitHub in the classroom

	Students	Instructors
Female	8.5%	6.9%
Male	90.0%	91.5%
Non-binary	1.0%	0.8%
Prefer to self-describe	0.5%	0.8%

Table 2: Student and instructor respondents' gender

The Results

GITHUB EDUCATION

Learning outcomes

To examine how implementing GitHub in the classroom predicts students' learning outcomes, we compared self-reported responses from students who used and students who did not use GitHub in the classroom during the previous semester. We also compared responses about students' experiences from teachers who implemented and from teachers who did not implement GitHub in the classroom. First, we examined how much students felt they learned about, and how much instructors felt their students learned about teamwork and collaboration, popular industry tool(s), and project management through their programming course. Separate analyses were conducted to examine teacher perceptions of students' learning outcomes. We used the Welch's t-test, accounting for unequal variance and unequal sample sizes.

In line with our hypothesis, we found that students who used GitHub in the classroom ($M = 3.65$, $SD = 1.30$) felt they learned more about teamwork and collaboration than students who did not use GitHub in the classroom ($M = 2.78$, $SD = 1.47$; $t(5743) = 25.60$, $p < 0.01$). Instructors who used GitHub in the classroom ($M = 3.12$, $SD = 1.44$) also felt their students learned more about teamwork and collaboration than students who did not use GitHub in the classroom ($M = 2.51$, $SD = 1.31$; $t(277) = 3.44$, $p < 0.01$). Results suggest that students who used GitHub in the classroom may have developed more skills necessary for collaborative work.

In addition, in line with our predictions, students who used GitHub in the classroom ($M = 3.72$, $SD = 1.14$) felt they learned more about popular industry tool(s) than students who did not use GitHub in the classroom ($M = 2.86$, $SD = 1.38$; $t(6032) = 28.01$, $p < 0.01$). Similarly, instructors who used GitHub in the classroom ($M = 3.78$,

SD = 1.04) felt their students learned more about popular industry tool(s) than students who did not use GitHub in the classroom (M = 3.24, SD = 1.20; $t(161) = 3.66$, $p < 0.01$). Thus, using GitHub in the classroom may lead students to feel better prepared for working in industry.

Lastly, students who used GitHub in the classroom (M = 3.53, SD = 1.23) felt they learned more about project management than students who did not use GitHub in the classroom (M = 2.68, SD = 1.38; $t(5697) = 26.54$, $p < 0.01$). In line with these results, instructors who used GitHub in the classroom (M = 2.95, SD = 1.15) felt their students learned more about project management than students who did not use GitHub in the classroom (M = 2.41, SD = 1.24; $t(276) = 3.62$, $p < 0.01$). These findings suggest using GitHub in the classroom may help students build skills important for tackling project objectives.

Preparation for the future

To examine how implementing GitHub in the classroom predicts students' feelings of preparation for the future, we compared self-reported responses about their own experiences from students who used and students who did not use GitHub in the classroom during the most recent semester. Again, we also compared responses about students' experiences from teachers who implemented and from teachers who did not implement GitHub in the classroom. We examined how much students felt their course prepared them, and how much instructors felt their course prepared their students for a future internship or career, being a part of the developer community, developing a portfolio of their work, and taking more advanced courses. Again, separate analyses were conducted to examine teacher perceptions of students' learning outcomes.

Consistent with our hypothesis, we found that students who used GitHub in the classroom (M = 3.84, SD = 1.12) felt more prepared for a future internship or career than students who did not use

GitHub in the classroom ($M = 3.25$, $SD = 1.30$; $t(5826) = 19.77$, $p < 0.01$). However, when we compared instructors who used ($M = 3.83$, $SD = 1.01$) versus did not use GitHub in the classroom ($M = 3.67$, $SD = 1.02$), we found no difference in their ratings of students' preparation for a future internship or career ($t(276) = 1.23$, $p = .22$). This suggests that, although instructors saw a connection between their course and students' future career regardless of using GitHub, students who used GitHub in the classroom found their courses more valuable to their future.

In addition, as predicted, we found that students who used GitHub in the classroom ($M = 3.68$, $SD = 1.22$) felt more prepared for being a part of the developer community than students who did not use GitHub in the classroom ($M = 2.82$, $SD = 1.38$; $t(5717) = 27.09$, $p < 0.01$). Similarly, instructors who used GitHub in the classroom ($M = 3.37$, $SD = 1.22$) felt their students were more prepared for being a part of the developer community than those who did not use GitHub in the classroom ($M = 2.75$, $SD = 1.20$; $t(276) = 3.99$, $p < 0.01$). Results suggest that students may feel more fit to contribute to the developer community after a semester of using GitHub in the classroom.

Supporting our predictions, we also found that students who used GitHub in the classroom ($M = 3.58$, $SD = 1.30$) felt more prepared for developing a portfolio of their work than students who did not use GitHub in the classroom ($M = 2.85$, $SD = 1.40$; $t(5516) = 21.66$, $p < 0.01$). In line with these results, instructors who used GitHub in the classroom ($M = 3.37$, $SD = 1.23$) also felt their students were more prepared for developing a portfolio of their work than students who did not use GitHub in the classroom ($M = 2.77$, $SD = 1.11$; $t(276) = 3.96$, $p < 0.01$).

Lastly, we found that students who used GitHub in the classroom ($M = 3.88$, $SD = 1.14$) felt more prepared for taking more advanced courses than students who did not use GitHub in the classroom ($M = 3.70$, $SD = 1.24$; $t(5529) = 5.97$, $p < 0.01$). However, instructors who did ($M = 3.84$, $SD = 1.07$) versus did not ($M = 4.05$, $SD = 0.91$) implement GitHub in the classroom showed no significant difference in ratings of students' preparedness for taking more

advanced courses (t(275) = 1.67, p = 0.10). This suggests, although instructors may see a connection between their course and more advanced courses regardless of using GitHub in the classroom, perhaps using GitHub in the classroom helps highlight that connection for students.

Classroom experiences

Next, to investigate how implementing GitHub in the classroom predicts students' classroom experiences, we compared responses from students who used and students who did not use GitHub in the classroom. Specifically, we focused on students' sense of belonging in the class and in the field. We also examined students' likelihood to recommend their course to others, and to take a similar course in the future. Because these questions centered on students' personal feelings and perceptions to which instructors may have less insight, we did not assess instructors' perceptions of how students felt.

Consistent with our hypothesis, we found that students who used GitHub in the classroom (M = 5.78, SD = 0.99) felt a greater sense of belonging in the classroom than students who did not use GitHub in the classroom (M = 5.49, SD = 1.12; t(5620.04) = 11.23, p < 0.01). In addition, students who used GitHub in the classroom (M = 5.49, SD = 1.26) also felt a greater sense of belonging in the field than students who did not use GitHub in the classroom (M = 5.05, SD = 1.52; t(5883) = 12.90, p < 0.01). Findings suggest that using GitHub in the classroom may help students feel like they are a part of a larger community of people, and that their contributions are valued.

In addition, as predicted, students who used GitHub in the classroom (M = 5.90, SD = 1.30) were more likely to recommend their course to others than students who did not use GitHub in the classroom (M = 5.50, SD = 1.58; F(1,34) = 3.55, p < 0.01). Students who used GitHub in the classroom (M = 5.78, SD = 1.43) also felt they were more likely to take a similar course in the future than

students who did not use GitHub in the classroom ($M = 5.43$, $SD = 1.70$; $t(5939) = 9.09$, $p < 0.01$). These results suggest students may enjoy and value their courses when they use GitHub in the classroom.

Design of implementation

Lastly, to explore how the design of implementation affects classroom learning, we conducted analyses focused on the way GitHub was used in the classroom. Running separate regression analyses, we found that the number of GitHub features used in the classroom predicted various student learning outcomes and feelings of preparation for the future. Specifically, using more GitHub features in the classroom predicted students learning more about teamwork and collaboration ($b = .17$, $t(2505) = 15.74$, $p < 0.01$), popular industry tools ($b = .11$, $t(2504) = 10.95$, $p = < 0.01$), and project management ($b = .16$, $t(2505) = 15.34$, $p < 0.01$). In addition, using more GitHub features in the classroom also predicted students feeling more prepared for a future internship and career ($b = .11$, $t(2501) = 11.40$, $p < 0.01$), being a part of the developer community ($b = .15$, $t(2500) = 14.78$, $p < 0.01$), developing a portfolio of their work ($b = .15$, $t(2499) = 13.65$, $p < 0.01$), and taking more advanced courses ($b = .09$, $t(2488) = 9.26$, $p < 0.01$). Consistent with our predictions, results suggest that it may not simply be the implementation of GitHub in the classroom that matters, but how GitHub is used in the classroom may be a key predictor of positive learning outcomes and experiences.

To further assess how the design of implementation affects classroom learning, we explored how feedback is perceived by students when it is delivered via GitHub. One way in which instructors use GitHub in the classroom is as a tool for feedback communication, and we examined students' perceptions of feedback effectiveness when feedback is delivered via GitHub versus not via GitHub. Of the students who used GitHub in the classroom and received instructor feedback, students who

received feedback via GitHub (M = 6.01, SD = 1.07) (versus not via GitHub (M = 5.72, SD = 1.34)) perceived feedback as more helpful ($t(790) = 4.43$, $p < 0.01$). Compared to students who did not receive feedback via GitHub (M = 5.58, SD = 1.40), students who did receive feedback via GitHub (M = 5.90, SD = 1.18) also felt their needs were better understood by their instructor ($t(733) = 4.48$, $p < 0.01$). Lastly, students who received feedback via GitHub (M = 5.99, SD = 1.13) felt they used the instructor's feedback more effectively than students who did not receive feedback via GitHub (M = 5.81, SD = 1.18, $t(666) = 2.67$, $p = .008$). However, there was no difference in students' understanding of instructor feedback ($t(1659) = 0.38$, $p = 0.70$) and the extent to which they pay attention to instructor feedback ($t(1668) = 1.04$, $p = 0.30$) when it was delivered via GitHub versus not via GitHub. In contrast, instructors reported no difference in feedback effectiveness across items when feedback is delivered via GitHub versus not via GitHub. Results suggest that student and teacher perceptions may not be in line with each other. Interestingly, delivering instructor feedback via GitHub seemed to impact students' perceptions that involved use or application of feedback, but not necessarily how much students paid attention to or understood feedback. One potential explanation is that the manner in which feedback is delivered may not shape students' willingness to attend to feedback or their ability to comprehend the feedback. However, how feedback is communicated may

influence its perceived value and thus impact the extent to which students feel understood by their instructors.

We also assessed students' perceptions of peer feedback effectiveness when feedback is delivered via GitHub versus not via GitHub. Of the students who used GitHub in the classroom and received peer feedback, students who received feedback via GitHub ($M = 5.96$, $SD = 1.10$) (versus not via GitHub ($M = 5.76$, $SD = 1.25$)) perceived feedback as more helpful ($t(682) = 2.50$, $p < 0.01$). Compared to students who did not receive peer feedback via GitHub ($M = 5.68$, $SD = 1.17$), students who received peer feedback via GitHub ($M = 5.99$, $SD = 1.03$) also felt they used the feedback more effectively ($t(687) = 4.14$, $p < 0.01$). However, there was no difference in students' understanding of peer feedback ($t(980) = 1.74$, $p = 0.08$), or in the extent to which students paid attention to peer feedback ($t(988) = 1.24$, $p = 0.22$) when it was delivered via GitHub versus not via GitHub. In contrast, instructors reported no difference in peer feedback effectiveness across items when feedback is delivered via GitHub versus not via GitHub.

The impact of GitHub in the classroom

Overall, we found that students who used GitHub in the classroom tended to have better student learning outcomes, feel more prepared for the future, and report more positive classroom experiences. In addition, we learned that although using GitHub in the classroom is an important predictor of learning outcomes, the design of implementation mattered. Thus, it is important to consider how students will interact with GitHub when implementing it in the classroom.

We found that students who used (versus did not use) GitHub in the classroom felt they learned more about teamwork and collaboration, popular industry tool(s), and project management via their programming course. These findings suggest students can build skills necessary for working with others in industry by using GitHub in the classroom. Collaborative experiences between students may make them more effective developers in industry, and it is likely that other platforms that facilitate similar experiences can also generate such benefits.

Next, we found that students who used GitHub in the classroom felt more prepared for their future career, being part of the developer community, developing a portfolio of their work, and taking more advanced courses. Research supports that one of the main motivations instructors have for implementing GitHub in the classroom is to help make students more competitive on the job market [3]. Based on our findings, students do perceive benefits in line with instructors' motivations to implement GitHub in the classroom. However, instructor and student findings were not completely in line with each other. Specifically, instructors reported no difference in student preparation for taking more advanced courses and for a future internship or career. One possible explanation is that students and teachers consider different sets of skills when evaluating student preparedness across domains, but further research is necessary to understand the discrepancies in responses.

In addition, our data supported that students who used GitHub in the classroom felt a greater sense of belonging in the classroom. One potential explanation is that using GitHub in the classroom allows students to collaborate and contribute to each other's learning process. Research finds that, more collaborative projects, where students work closely with peers, can foster a stronger sense of belonging [7]. Importantly, students' sense of belonging can positively predict variables tied to academic success, such as intrinsic motivation and academic self-efficacy [4]. We also found

that using GitHub in the classroom may promote a greater sense of belonging in the field more generally. One potential explanation is that GitHub supports interactions with individuals in industry. For instance, in a previous study, the researchers reported that unregistered students and other GitHub users visited and "starred" their public repositories, establishing a connection between students and others in the field [15].

Lastly, we learned that the design of implementation may impact student's learning outcomes and preparation for the future. For instance, using more features within GitHub may predict greater student learning outcomes. Because GitHub can be flexibly used in the classroom for tasks ranging from distributing classroom content to providing line-by-line feedback via pull requests, it is unsurprising that engaging more deeply with GitHub may boost benefits associated with using it. Further, we uncovered that receiving instructor and peer feedback via GitHub may have boosted students' perceptions of the feedback's value. In particular, using GitHub in the classroom predicted students' effective use of feedback and perception of feedback as helpful, but not students' attention towards or understanding of feedback. Importantly, students who received instructor feedback via GitHub felt better understood by their instructor. These findings highlight future avenues for research. For example, from this survey, we did not learn how students are actually applying instructor feedback when it is delivered via GitHub or via other means. It may be the case that receiving feedback via GitHub improved students' perceptions of the feedback but did not impact how students actually applied the feedback. However, it may also be the case that students are better able to apply feedback when it is delivered via GitHub versus via other means. Developing a deeper understanding of how students perceive and apply

feedback delivered via different means will help instructors leverage tools that enhance their teaching. Lastly, it is unclear why instructors' responses were not in line with those of students. One possibility is that instructors lack insight to students' more private experiences that meaningfully shift relative to the use of GitHub in the classroom to deliver feedback.

Limitations of the research

Our research is a step towards understanding the impact associated with using GitHub in the classroom. However, there are a few notable limitations of the research that should be considered. First, our student and instructor participants may have been more experienced with using GitHub than the average user of GitHub because we only recruited participants who had verified student or instructor status through GitHub. It is possible that the average student would benefit even more from using GitHub in the classroom than our participants because they were learning a valuable industry tool with which they were less familiar. On the other hand, it is possible that the average student would benefit less from using GitHub in the classroom than our participants because they may be less committed to learning GitHub. To improve the generalizability of our findings, future research should attempt to replicate our pattern of results using a sample more representative of the average student using GitHub in the classroom.

In addition, the replication of findings is especially important because our effects are small. Although it may be the case that the benefits associated with using GitHub are small, another

reason could be that we recruited a diverse sample of participants who used GitHub in various types of classrooms. Thus, it is possible that examining the effect of using GitHub in a more controlled classroom study would produce larger effects. Although we cannot be fully confident in these results without replication and further research, the pattern of results was mostly consistent across variables. Overall, our findings support that using GitHub in the classroom is associated with positive student learning outcomes and experiences.

Lastly, our study design did not allow us to draw conclusions regarding causality. Although we could determine how students who used (versus did not use) GitHub in the classroom differed, our correlational data did not reveal how using GitHub in the classroom caused change in students' learning outcomes and experiences. So, although we could examine the relationship between key variables and make inferences about causality, we did not collect data that directly examined causality. Future research may use the experimental method to explore whether implementing GitHub in the classroom causes change in students' learning outcomes and experiences. For example, researchers may randomly assign classes to either implement or not implement GitHub in the classroom, and observe how the two conditions differ on key variables.

Why use GitHub in your classroom

Our research provides preliminary evidence illustrating how using GitHub in the classroom may benefit student learning and classroom experiences. Results highlight the potential impact of using GitHub in the classroom on student learning outcomes, and suggests using GitHub in the classroom positively predicts learning outcome variables. Importantly, the design of GitHub implementation matters and our results support that how students use GitHub in the classroom may impact learning outcomes. Combined with further research, our findings may be used to inform best practices for implementing GitHub in the classroom.

References

[1] Butler, D.L., & Winne, P.H. (1995). Feedback and self-regulated learning: A theoretical synthesis. Review of Educational Research, 65, 245-281.

[2] Falkner, K., & Falkner, N.J.G. (2012). Supporting and structuring “contributing student pedagogy” in computer science curricula. Computer Science Education, 22, 413-443.

[3] Feliciano, J., Storey, M., & Zagalsky, A., (2016). Student experiences using GitHub in software engineering courses: A case study. ICSE ‘16 Companion, 1- 10.

[4] Freeman, T.M., Anderman, L.H., & Jensen, J.M. (2007). Sense of belonging in college freshmen at the classroom and campus levels. Journal of Experimental Education. 75, 203-220.

[5] GitHub. (n.d.). Retrieved August 1, 2018, from http://github.com

[6] Gunnarsson, S., Larsson, P., Mansson, S., Martensson, E., & Sonnerup, J. (2017). Enhancing student engagement using GitHub as an educational tool. Introduction to Teaching and Learning in Higher Education. 1-14.

[7] Kember, D., Lee, K., & Li, N. (2001). Cultivating a sense of belonging in parttime students. International Journal of Lifelong Education. 20, 326-341.

[8] Kertész, C. (2015). Using GitHub in the classroom - a collaborative learning experience. International Symposium for Design and Technology in Electronic Packaging, Brașov, Romania, 381-386.

[9] Marcketti, S.B., & Karpova, E.E. (2014). Getting ready for the real world: Student perspectives on bringing industry collaboration into the classroom. Journal of Family and Consumer Sciences, 106, 27-31.

[10] O’Keefe, P. (2013). A sense of belonging: Improving student retention. College Student Journal. 47, 605-613.

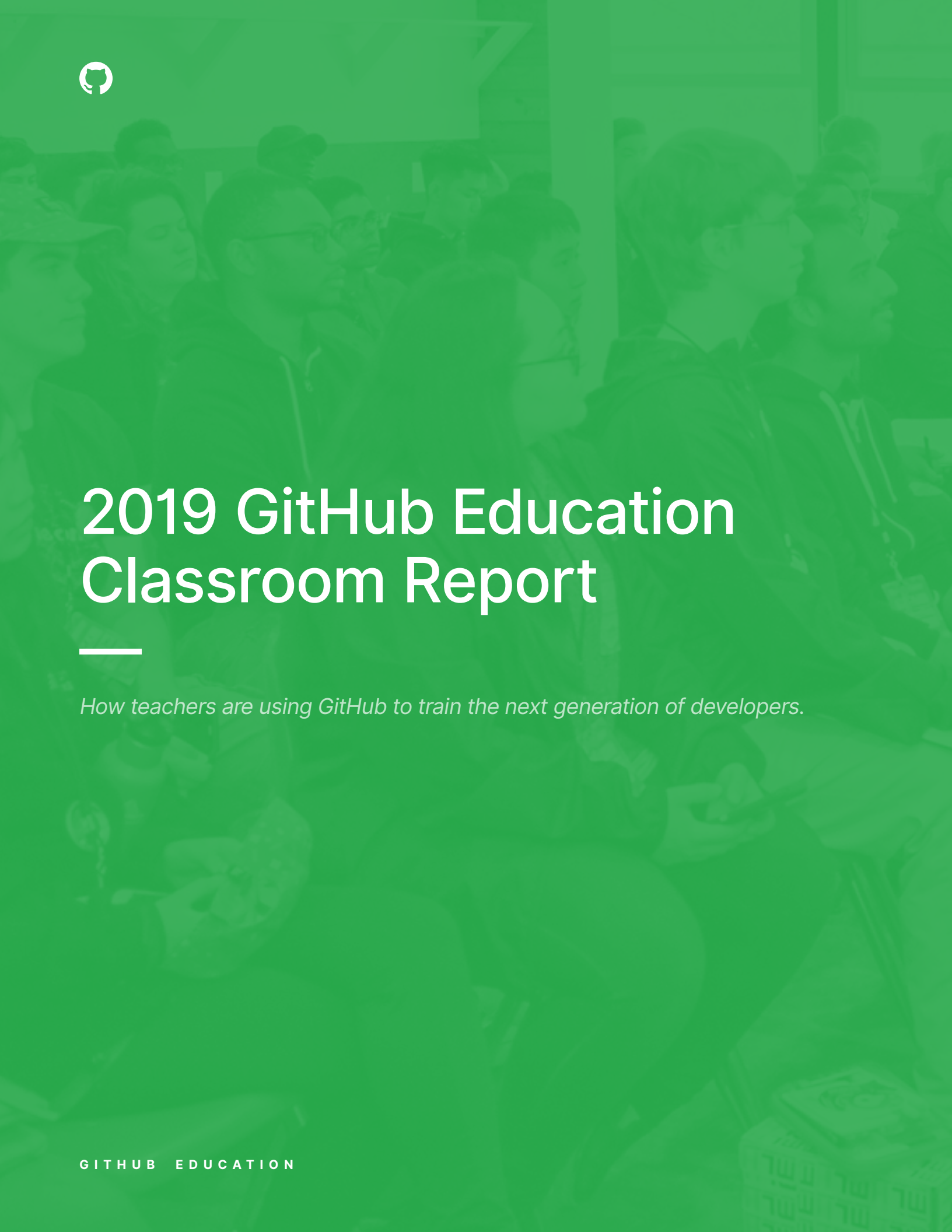
[11] Pajares, F. (1996). Self-efficacy beliefs in academic settings. Review of Educational Research, 66, 543-578.

[12] Trujillo, G., & Tanner, K.D. (2014). Considering the role of affect in learning: Monitoring students’ self-efficacy, sense of belonging, and science identity. CBE – Life Sciences Education, 13, 6-15.

[13] Walton, G.M., Cohen, G.L., Cwir, D., & Spencer, S.J. (2012). Mere belonging: The power of social connections. Journal of Personality and Social Psychology, 102, 513-532.

[14] Wan, T. (2018, June, 19). GitHub’s new education bundle equips students with industry-standard coding tools. Retrieved from https://www.edsurge.com

[15] Zagalsky, A., Feliciano, J., Storey, M.A., Zhao, Y., & Wang, W. (2015). The emergence of GitHub as a collaborative platform for education. In Proc. of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 1906-1917.



2019 GitHub Education Classroom Report

How teachers are using GitHub to train the next generation of developers.

GITHUB EDUCATION

Report summary

Each year, we survey students and teachers to find out how they're using GitHub and how their classroom experience is changing. In 2019, we surveyed nearly 16,000 students and over 100 educators to find out not just what text editor they use, but how GitHub fits into their classrooms and careers.

We ask these questions to inform the entire technical education community: To help students see what their futures might look like, to help teachers make more effective materials, and to help companies like ours decide what products to build. Through this project, the GitHub Education team can contribute to the body of insights on learning and technical education—like when Vanessa Gennarelli explained at SIGCSE 2019 that "the greater number of GitHub features used, the greater the positive learning outcomes."

Listening to students and teachers—drawing from their first-hand experiences—helped the GitHub Education team decide to build integrations between GitHub Classroom and learning management systems, and develop improvements to the GitHub Campus Experts program, to name just two examples.

With these goals in mind, we asked questions that would inform us about how students use the tools available to them, how they choose those tools, and what kind of work they've gone on to accomplish. We asked teachers about their classrooms, how they use GitHub, and what their plans are for the future.

What we learned

We reached out to students and teachers who have benefited from GitHub Education, and 15,941 students and 108 teachers responded. If you can only know three things about what we learned, here are the takeaways:

Students and teachers love using GitHub

We asked students and teachers, “How likely is it that you would recommend GitHub to a friend or colleague?” to calculate GitHub’s Net Promoter Score (NPS). Students and teachers agreed: GitHub is worth recommending.

NPS is an industry measure of customer sentiment that compares the percentage of strong supporters, known as promoters, to the percentage of strong critics, known as detractors. NPS ranges from -100 (all detractors) through zero (evenly split) to 100 (all promoters).

Among students, GitHub has a high NPS of 63.

Net promoter score: Student survey

How likely is it that you would recommend GitHub to a friend or colleague?

	Overall	Current users	Users who quit	Used GitHub in classroom	Workplace users
NPS	63	69	24	73	74
Detractors	6%	3%	19%	2%	2%
Passives	26%	24%	38%	22%	21%
Promoters	69%	72%	43%	76%	77%

Among teachers, GitHub also has a high NPS of 60.

Net promoter score: Teacher survey

How likely is it that you would recommend GitHub to a friend or colleague?

	Overall	Current users	Implements GitHub in class	Uses GitHub Classroom
NPS	60	67	70	76
Detractors	7%	3%	5%	5%
Passives	27%	27%	21%	14%
Promoters	66%	70%	74%	81%

Teachers are equipping students with essential career skills

With GitHub, teachers bring industry-standard programming languages and version control into their courses. Teachers introduce languages that have wide use in the industry and teach the near-universal Git skills that benefit students long after they leave the classroom.

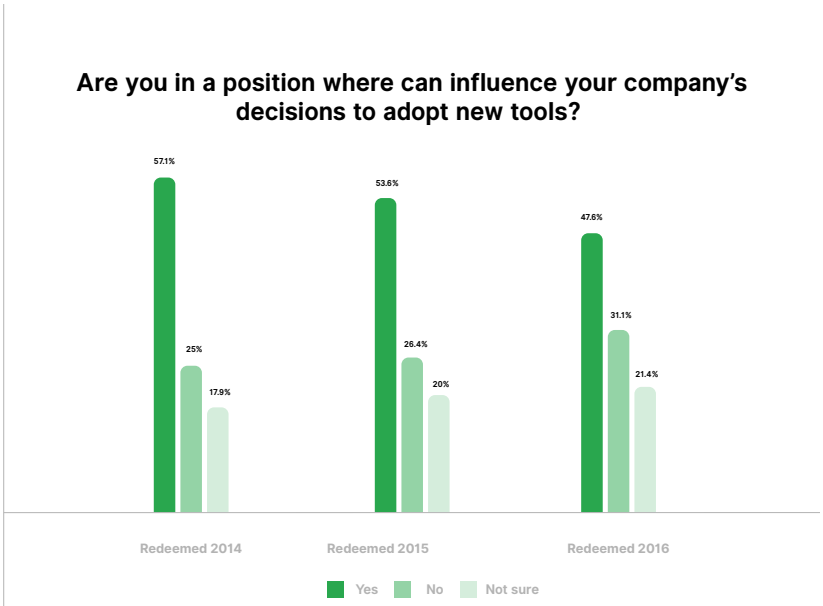
Students who use GitHub grow into technology careers

When we looked specifically at students who joined our programs in 2014 and 2015, we found that a large portion of those students have gone on to careers in software development, design, and product management, among others. On top of that, they're

bringing fresh perspectives and experience with tools into the workplace. For those who joined in 2014, over half reported having an influence on tool selection.

Influence on adoption of new tools by year of student coupon redemption

Respondents who redeemed student coupons in 2014 were more likely to be in a position to influence decisions to adopt new tools than those who redeemed in 2015 or 2016



Who we surveyed

To get in touch with people for the survey, we emailed students who joined GitHub's student programs in 2014, 2015, and 2016, and emailed teachers who joined our educational programs in 2014 and 2016. Participating students and teachers received different surveys with questions specific to their experiences.

In each survey, we collected some demographic information, such as gender identity and age, and for students specifically, race, ethnicity, and their highest level of education.

Student and teacher respondents mostly identified as male.

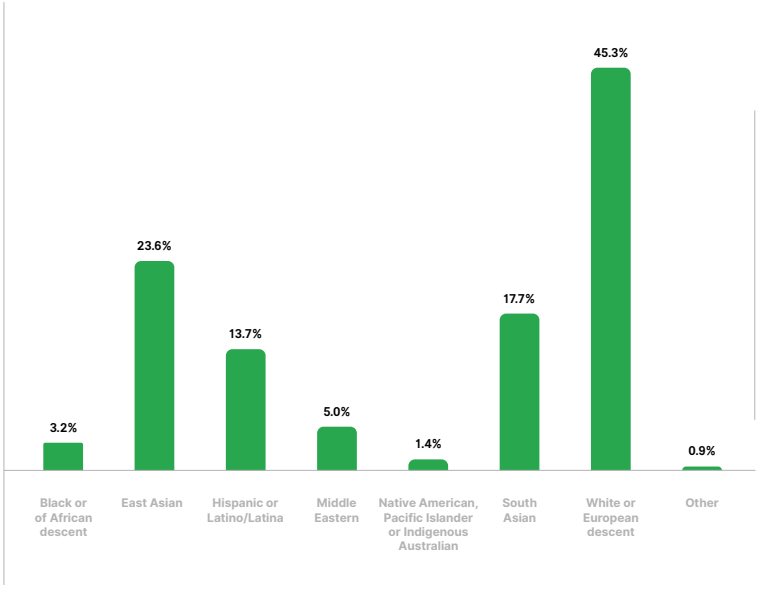
Identification

	Female	Male	Non-binary	Self-described
Students	11.2%	87.4%	1.0%	0.4%
Teachers	6.9%	93.1%	0%	0%

The average age of student respondents was 26.5 years old, while teachers averaged 41.3 years old.

From students, we learned a bit more about their race and ethnicity by asking them to choose all that applied:

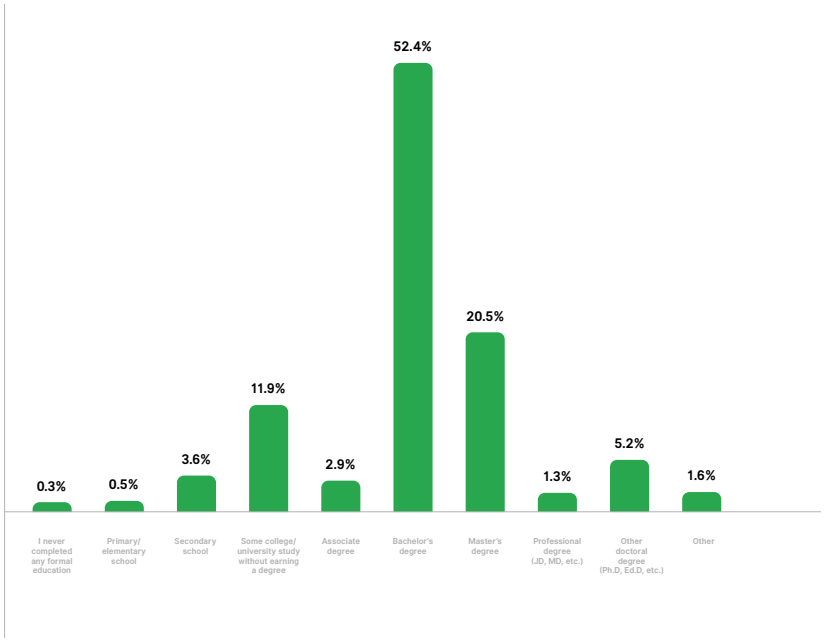
Race and Ethnicity



We also asked students to tell us about their highest level of education. About half of respondents reported to have a

bachelor's degree and nearly all completed at least some university-level study.

Highest Level of Education



We'll cover all three of our takeaways in more detail below, starting with students.

GitHub students become valued professionals

Through our survey, we learned that students aren't just learning how to use powerful tools for their coursework. Students are setting themselves up for success in the workplace.

Earlier this year, we interviewed Spencer Kaiser, Principal Architect of Emerging Technology at American Airlines. Spencer told us how, as a student, GitHub and the GitHub Student Developer Pack gave him the real-world tools to develop his skills beyond the confines of a classroom and textbook. And with those skills,

he was able to level up and become a workplace leader. As he told us, “I’m responsible for identifying potential technology that American Airlines could adopt, either software or hardware infrastructure.”

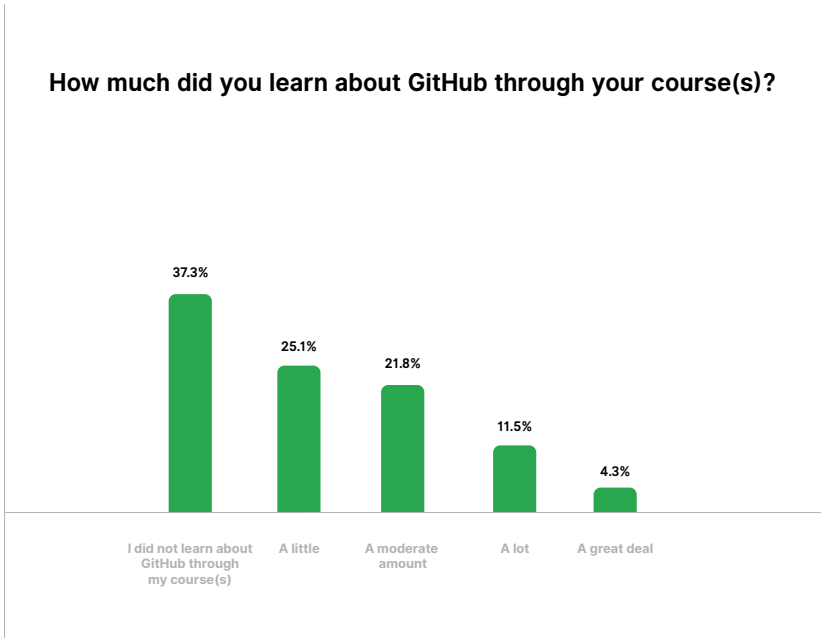
Spencer’s story is an illustration of a progression reflected in the survey data: GitHub users grow from students and independent learners into teammates and leaders with valued perspectives.

It starts in the classroom

Let’s take a look at how students are using GitHub while in school. For starters, GitHub is showing up to class: Over half of the students surveyed reported learning at least a little about GitHub through their coursework, though a plurality (37.3 percent) noted that this was missing from their curriculum.

GitHub in the classroom

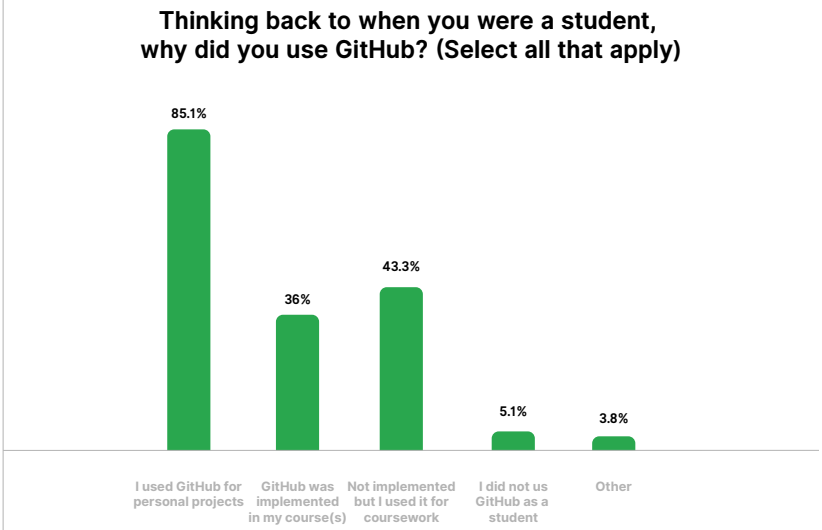
Over half of respondents who redeemed student coupons in 2014-2016 learned at least a little about GitHub through their course(s)



Not only are students learning about GitHub in class, but they’re using it on their own. Over 40 percent of respondents said they used GitHub for coursework, even when it wasn’t part of the course itself. And 85.1 percent of respondents said that they used GitHub for a personal project.

Use of GitHub during school

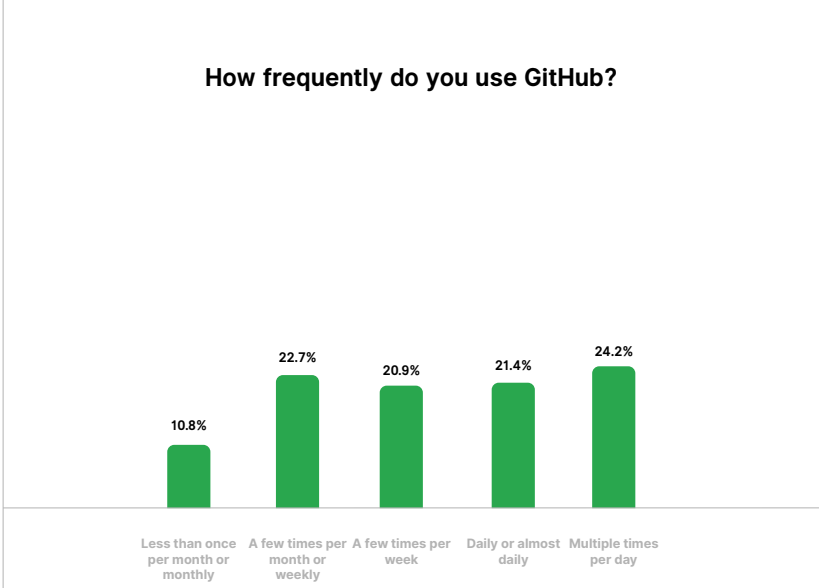
Most respondents who redeemed student coupons in 2014-2016 used GitHub during school for personal projects



GitHub use among students isn't an isolated once-a-semester event, either. The overwhelming majority of student respondents said that they use GitHub routinely. Nearly half the students reported using GitHub every day, with some using it multiple times a day. Students aren't leaving it behind when they graduate, either. In fact, they're finding the rewards that come with becoming GitHub power users.

Frequency of use

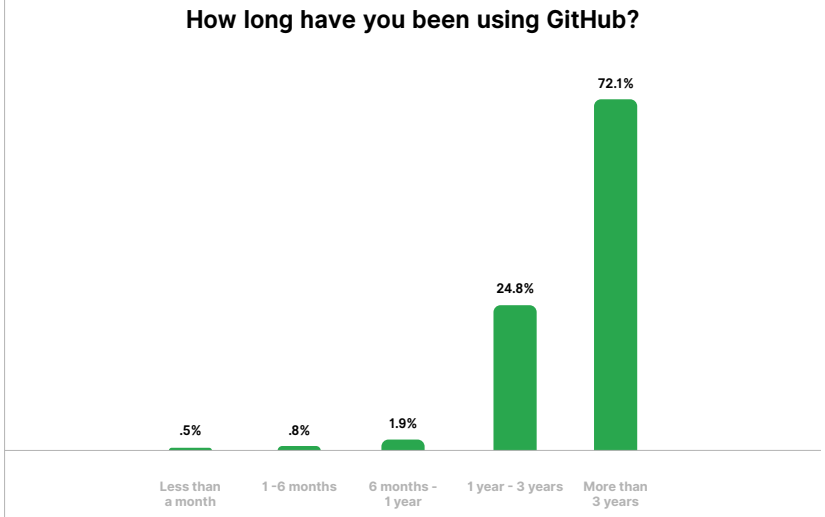
Most respondents who redeemed student coupons in 2014-2016 use GitHub at least a few times per week



What's more, students find that they're sticking with GitHub. Most respondents have been using GitHub for a long time, more than three years.

Time as a GitHub user

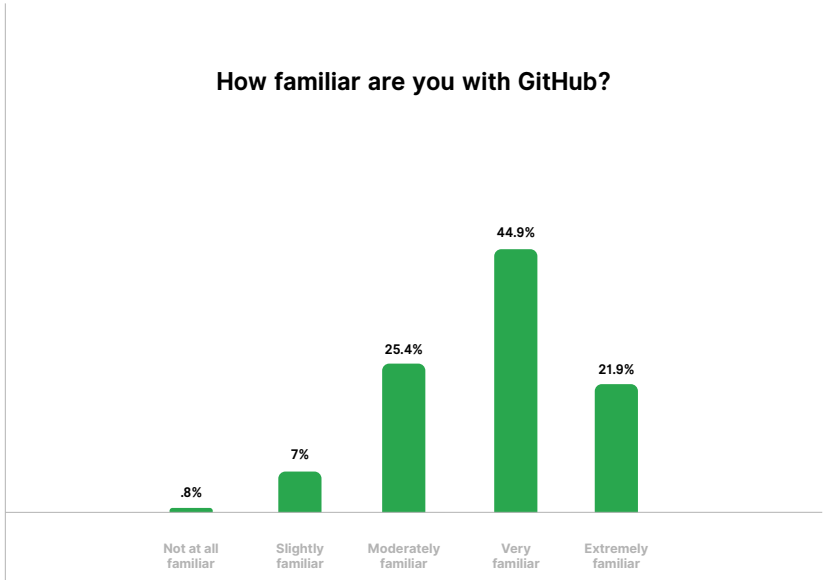
Most respondents who redeemed student coupons in 2014-2016 have been using GitHub for more than 3 years



And sticking with GitHub has paid off: A majority of respondents said they consider themselves very familiar with GitHub.

Familiarity with GitHub

Over half of respondents who redeemed student coupons in 2014-2016 consider themselves very familiar to extremely familiar with GitHub

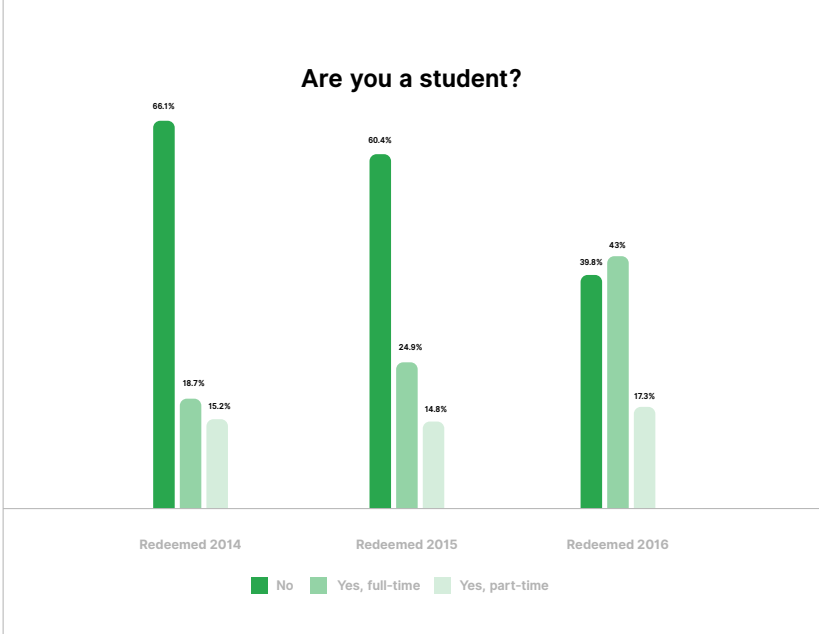


Preparing students for the workplace

After classes and internships, what do students have to look forward to? Students who became members of our programs in 2014 and 2015 have largely moved on from their studies and into the workplace, giving us an idea of what's to come for today's students.

Student status by year of student coupon redemption

Over half of respondents who redeemed their student coupon in 2014 and 2015 are currently not students

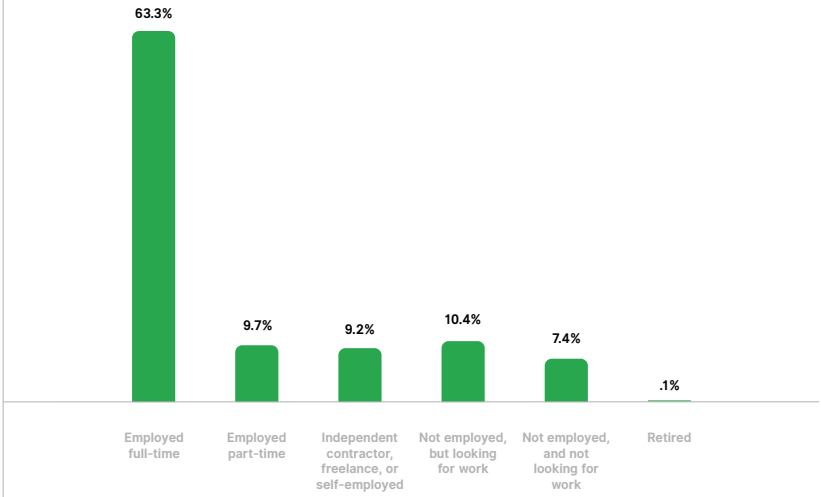


A majority of the respondents reported being employed full time. Though developer jobs in the information technology industry are common, that doesn't tell the whole story. The students in our survey have gone on to a broad range of roles, from developer to product manager to designer to business analyst, and they're bringing their skills to a broad range of industries.

Employment status

Over half of respondents who redeemed student coupons in 2014-2016 are employed full-time

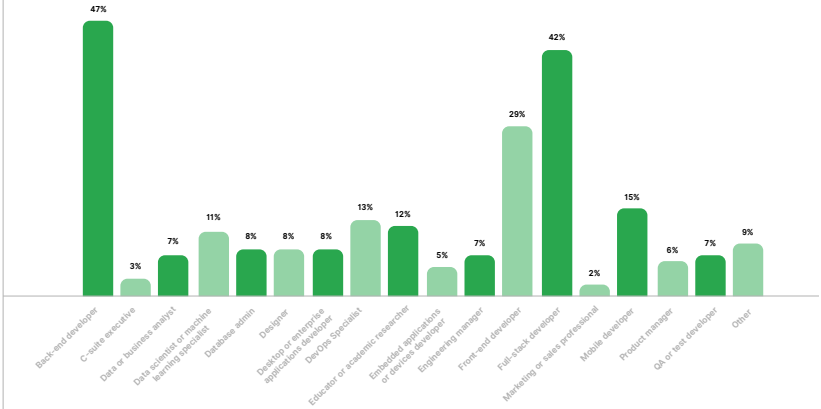
What is your employment status?



Role

Back-end developer, full-stack developer, and front-end developer are the most common roles among respondents who redeemed student coupons in 2014-2016

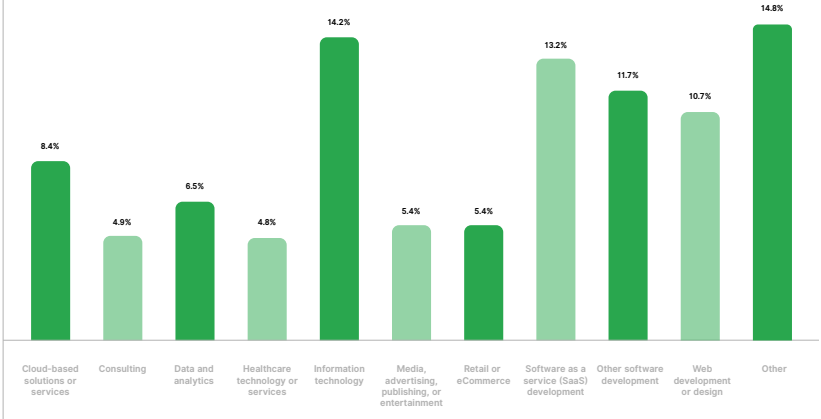
Which of the following roles describe you? (Select all that apply)



Industry

Information technology, and Software as a service (SaaS) development are the most common industries to work in among respondents who redeemed student coupons in 2014-2016

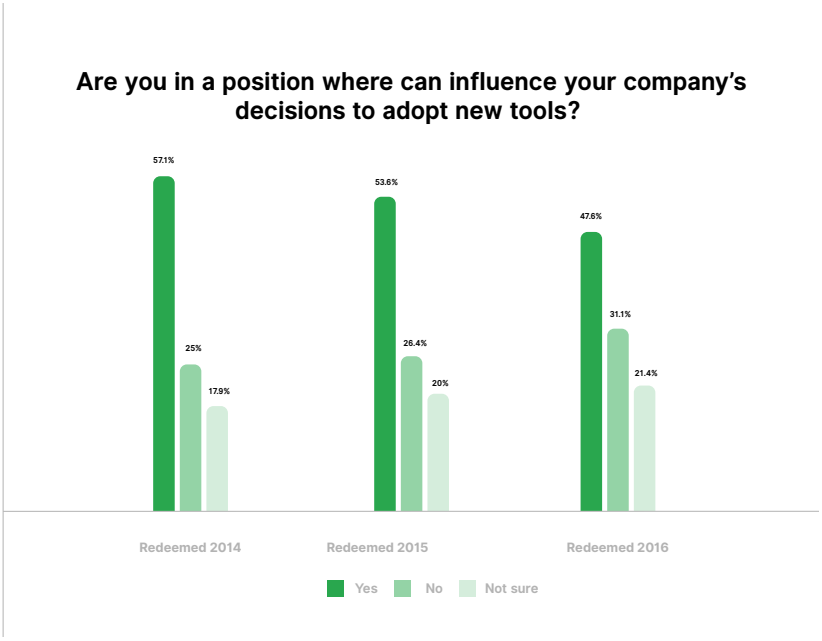
In which industry do you work?



Whatever their industry, GitHub students aren't just coasting along. It appears that, as students grow into their roles, they share their perspectives on the tools used in their workplace. Students who joined us in 2014 were most likely to say yes when asked, "Are you in a position where you can influence your company's decisions to adopt new tools?"

Influence on adoption of new tools by year of student coupon redemption

Respondents who redeemed student coupons in 2014 were more likely to be in a position to influence decisions to adopt new tools than those who redeemed in 2015 or 2016

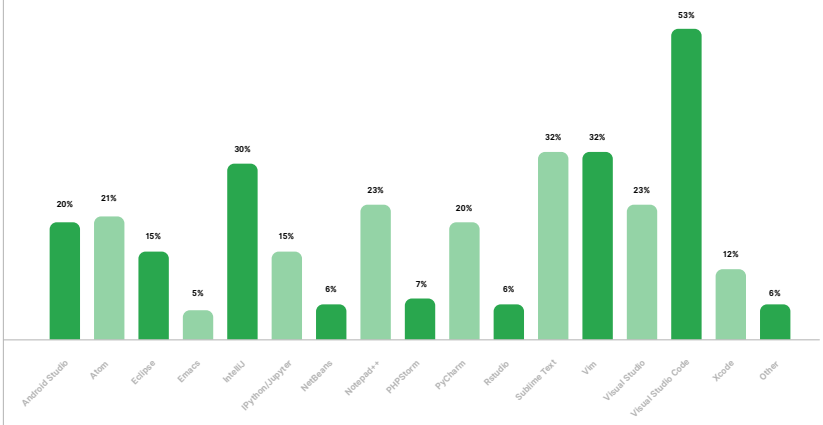


When it comes to choosing tools, survey data also shows that the students who joined GitHub programs go on to use a wide range of tools in the workplace. Through the survey, we took a snapshot of what tools they use. It's not just about version control: They're using office suites, issue trackers, IDEs, programming languages, and more. This data helps illustrate the trends in what tools junior developers use and gives current students an idea of what tools are available to them.

Integrated development environments

Visual Studio Code, Sublime Text, Vim, and IntelliJ are among the most commonly used IDEs by respondents who redeemed student coupons in 2014-2016

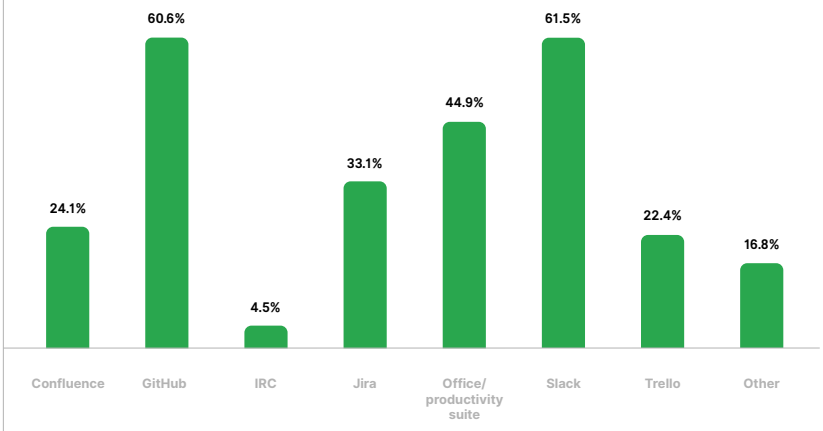
Which of the following integrated development environments (IDEs) do you typically use? (Select all that apply)



Communication tools

Over half of the respondents who redeemed student coupons in 2014-2016 use Slack and GitHub to communicate, coordinate, or share knowledge with coworkers or collaborators

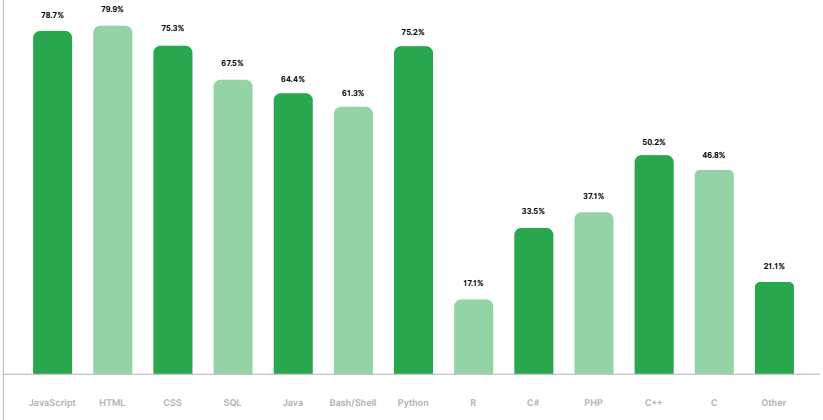
Which of the following tools do you use to communicate, coordinate, or share knowledge with your coworkers or collaborators? (Select all that apply)



Programming languages

HTML, JavaScript, CSS, and Python are among the most commonly used programming languages by respondents who redeemed student coupons in 2014-2016

Which of the following programming language(s) have you used? (Select all that apply)



Taken as a whole, the survey responses show that GitHub is an important part of many students' growth not just as learners and developers—but as professionals.

If you're a student learning GitHub in the classroom or on your own, then get ahead and sign up for the GitHub Student Developer Pack, which gives you free access to dozens of the best tools for developers.

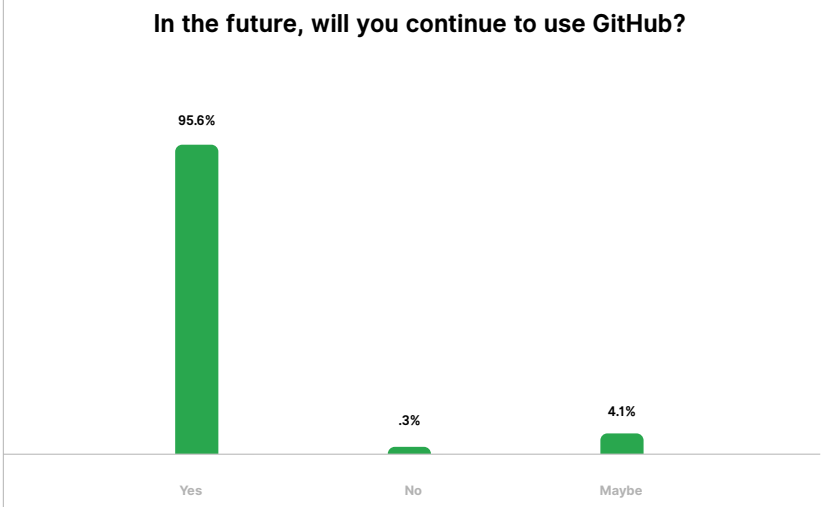
GitHub students learn life skills, not just job skills

Survey responses show that GitHub students aren't using GitHub because it's a requirement. They're folding GitHub into their lives beyond the classroom and workplace.

Survey responses from students make it clear that students want to use GitHub. Nearly all current GitHub users planned to continue to use it in the future (95.6 percent). Of the respondents who don't currently use GitHub, nearly all either want to use it in the future (46.6 percent) or are considering it (47.7 percent).

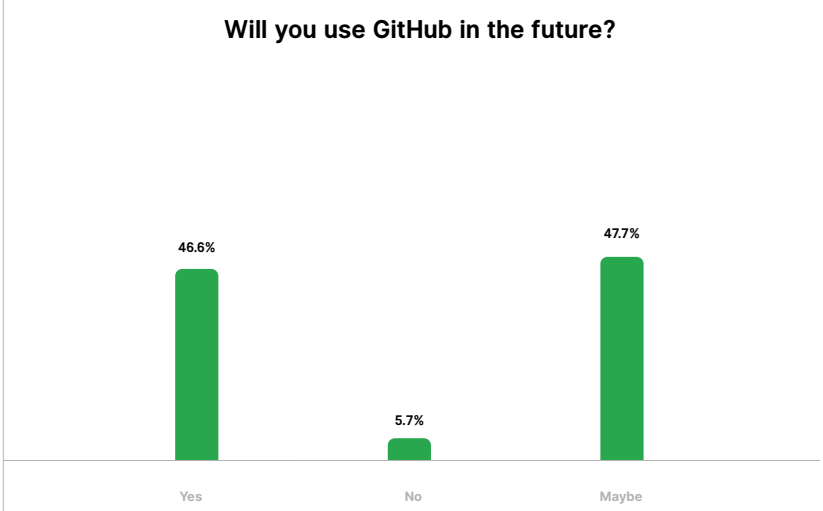
Current users of GitHub: Continue to use GitHub in the future?

Nearly all of the respondents who redeemed student coupons in 2014-2016 and currently use GitHub felt they would continue to use GitHub in the future



Non-users of GitHub: Use GitHub in the future?

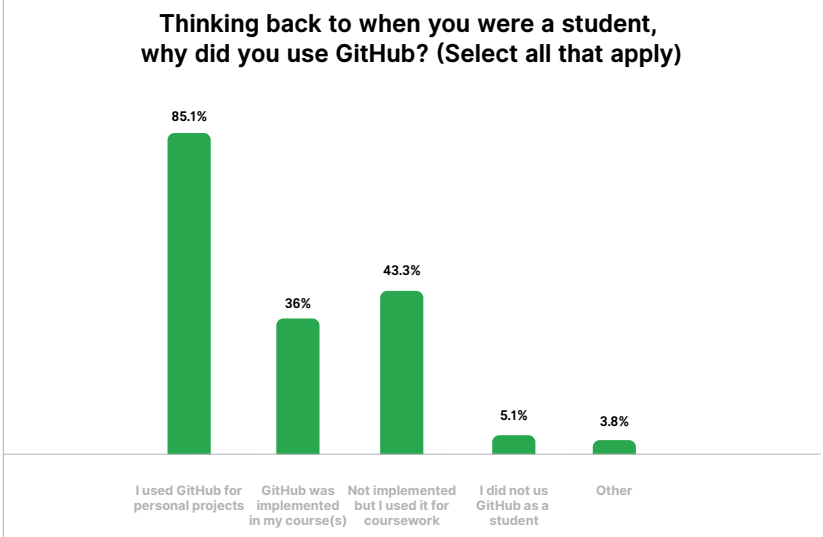
Nearly half of the respondents who redeemed student coupons in 2014-2016 but are currently non-users of GitHub felt they would use GitHub in the future



Students see uses for GitHub in their studies beyond course requirements. Nearly half of students (43.3 percent) said that they used GitHub for coursework that did not require GitHub. And an overwhelming majority of students who responded to the survey said they used it for personal projects.

Use of GitHub during school

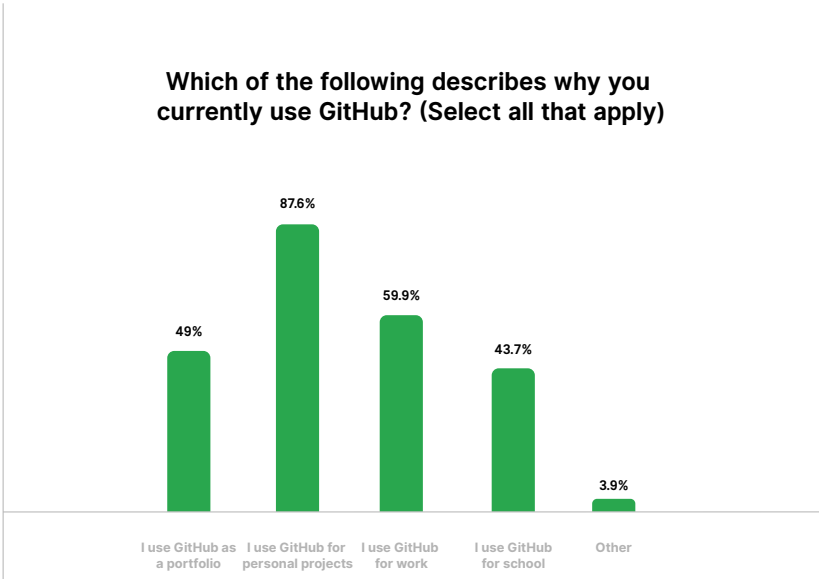
Most respondents who redeemed student coupons in 2014-2016 used GitHub during school for personal projects



Beyond that, nearly half the people who joined GitHub student programs (49 percent) noted using GitHub as a portfolio for the work that they'd done. To learn more about GitHub as a portfolio, check out the #PinstoWin competition on Twitter, which highlights how current students are showing off their work with GitHub.

Why use GitHub

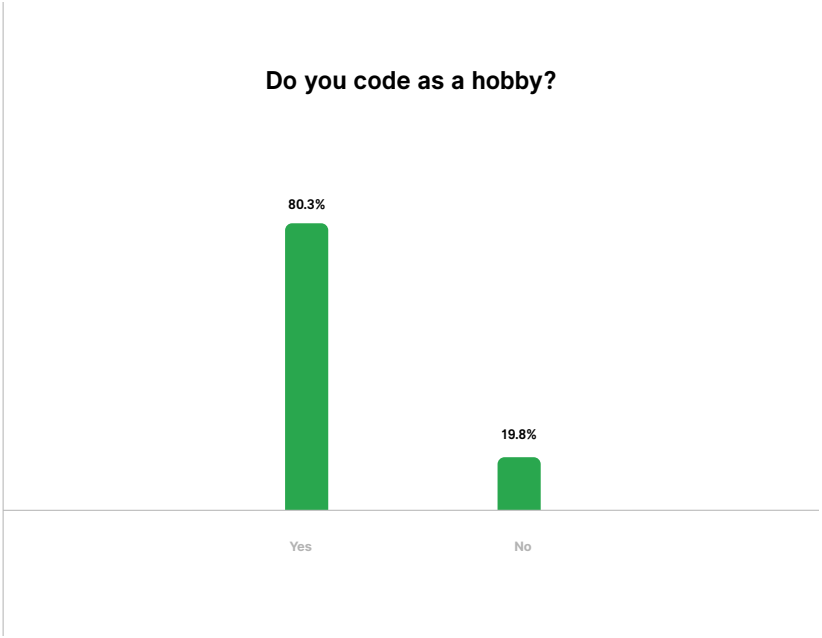
Over half of respondents who redeemed student coupons in 2014-2016 use GitHub for personal projects and for work



Of course, all work and no play makes a dull Octocat. Students widely reported that they find enjoyment in coding as a hobby, too.

Coding as a hobby

Most respondents who redeemed student coupons in 2014-2016 code as a hobby



GitHub teachers equip students with essential skills

Through our survey, we learned how teachers are preparing students to succeed in both their classes and careers, and how they're using GitHub to do it.

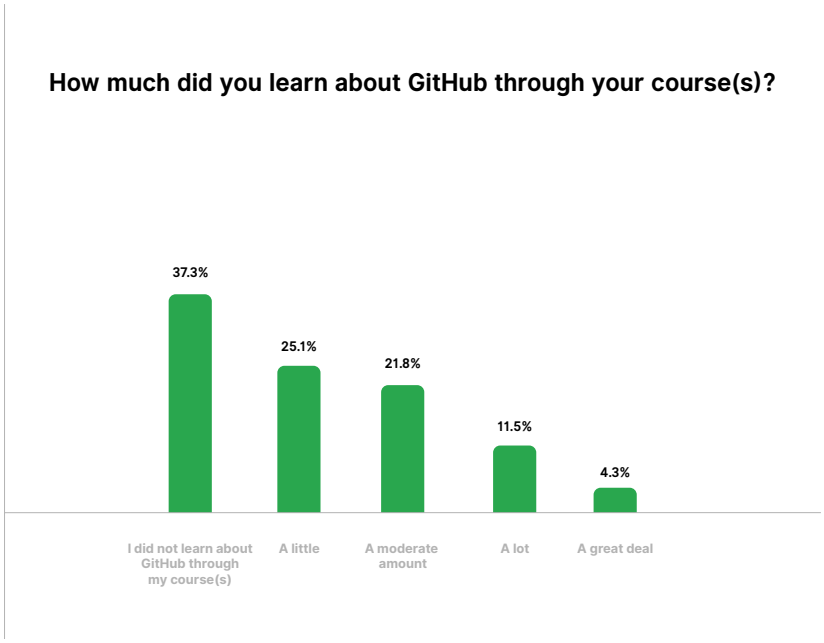
Whether or not teachers use GitHub in class, they have a big impact on what tools students use and what skills they develop, and that impact extends into students' careers in the industry. For an illustration, look no further than our interview with Spencer Kaiser (reminder that he's the Principal Architect of Emerging Technology at American Airlines). Spencer told us how, with just

a nudge from one of his professors, GitHub and the Student Developer Pack gave Spencer the real-world tools to develop his skills on his own. The survey data shows that many students are having experiences like Spencer's.

As we mentioned earlier, in response to our question on using GitHub in the classroom, over half of the students surveyed reported learning at least a little about GitHub through their coursework.

GitHub in the classroom

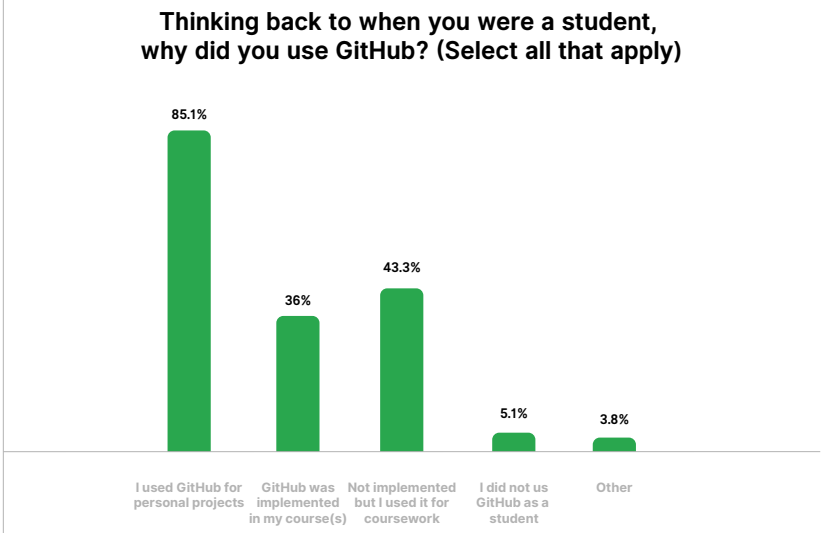
Over half of respondents who redeemed student coupons in 2014-2016 learned at least a little about GitHub through their course(s)



But even if teachers don't choose to bring GitHub into their classrooms, students aren't ignoring it. When we asked about using GitHub during school, over 40 percent of the respondents said they used GitHub for coursework, even when it wasn't part of the course itself.

Use of GitHub during school

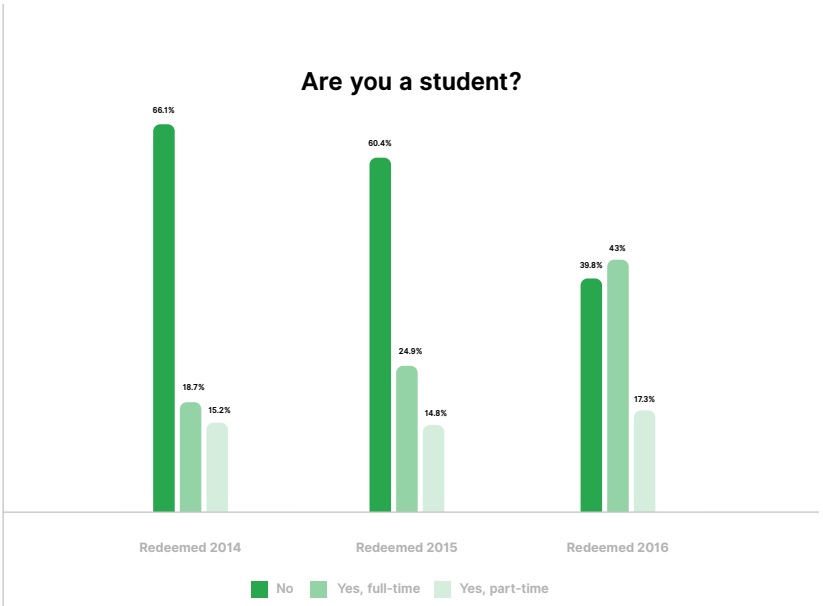
Most respondents who redeemed student coupons in 2014-2016 used GitHub during school for personal projects



By looking at the students who joined GitHub student programs in 2014 and 2015 and have largely moved on from their studies into the workplace, we can see what's in store for current students.

Student status by year of student coupon redemption

Over half of respondents who redeemed their student coupon in 2014 and 2015 are currently not students

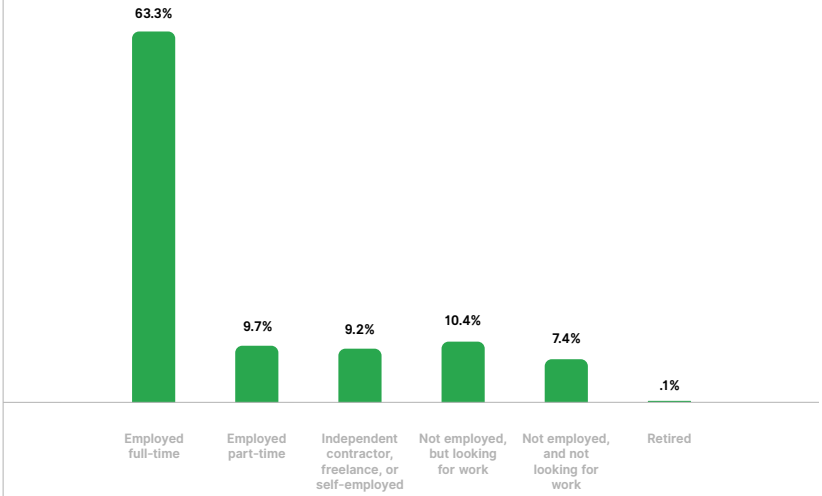


If past GitHub students are any indication, students are headed toward careers that make use of the skills they acquire in the classroom. Students in the survey have gone on to a broad range of roles, from developer to product manager to designer to business analyst and they're bringing their skills to a broad range of industries.

Employment status

Over half of respondents who redeemed student coupons in 2014-2016 are employed full-time

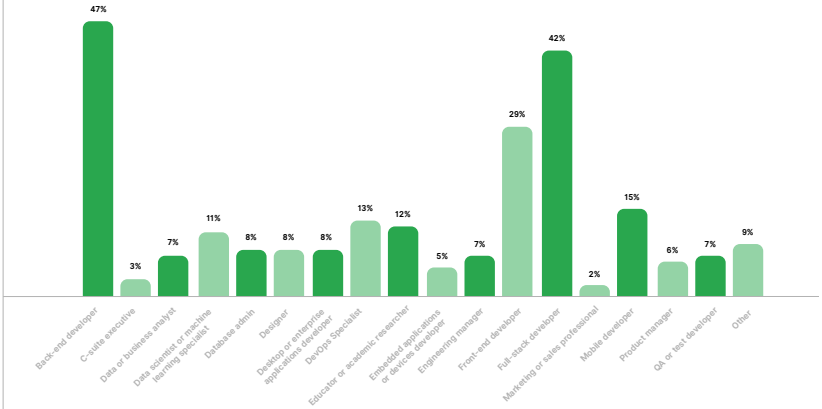
What is your employment status?



Role

Back-end developer, full-stack developer, and front-end developer are the most common roles among respondents who redeemed student coupons in 2014-2016

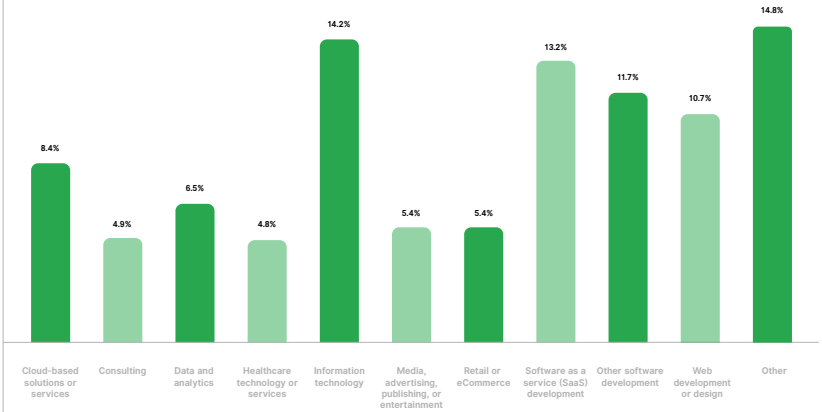
Which of the following roles describe you? (Select all that apply)



Industry

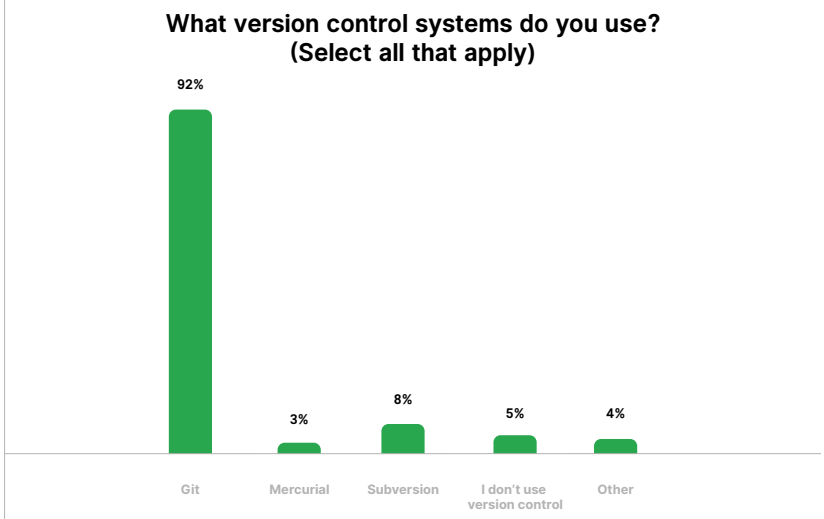
Information technology, and Software as a service (SaaS) development are the most common industries to work in among respondents who redeemed student coupons in 2014-2016

In which industry do you work?



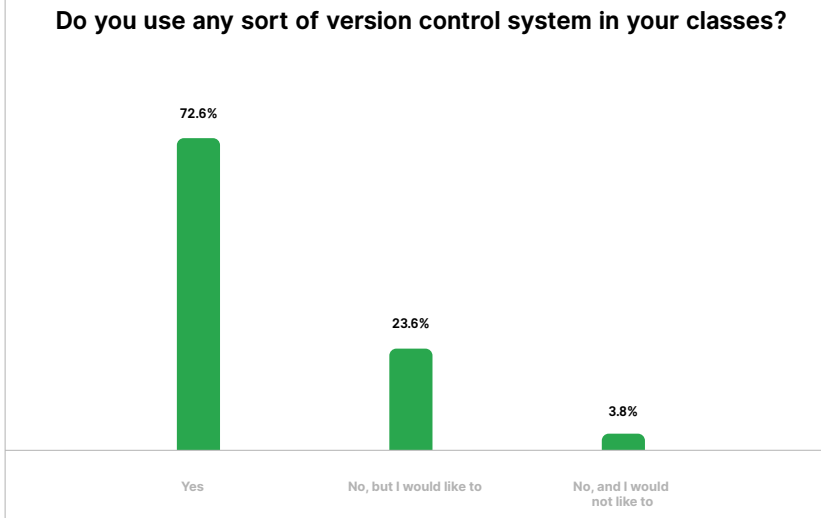
Survey data also shows that the students who joined go on to use a wide range of tools in the workplace: office suites, issue trackers, IDEs, programming languages, and version control, just to name a few. When it comes to version control, Git has become a de facto standard.

Version control systems



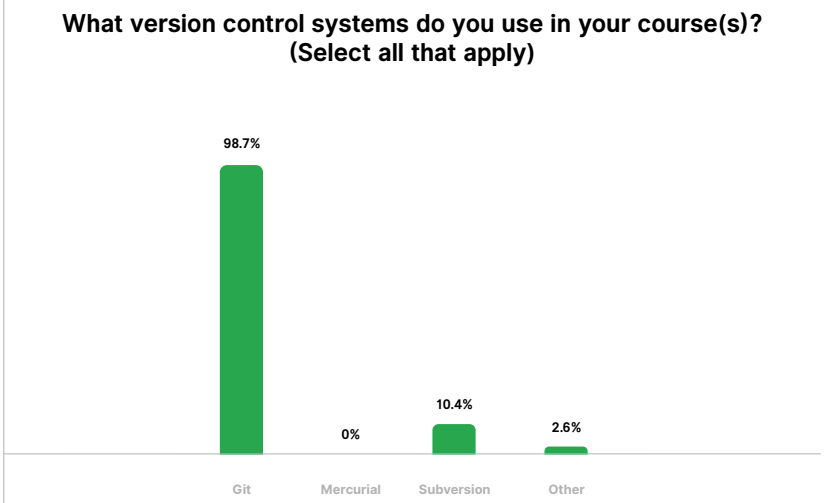
Use of version control

Nearly three-quarters of teacher survey respondents use some sort of version control system in their classes



Use of version control

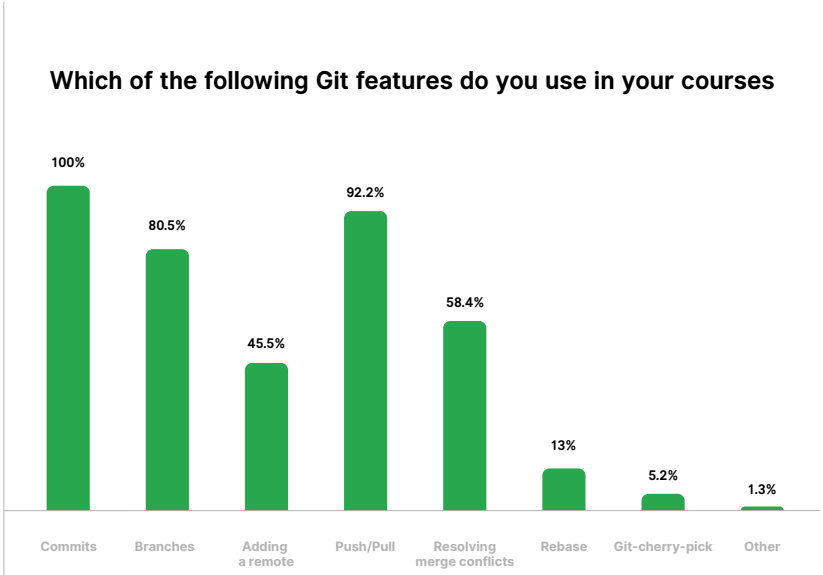
Of the teacher survey respondents who use some sort of version control system in their classes, nearly all of them use git



On top of that, coverage of essential features of Git, such as commits, branches, and pushing and pulling, are being covered in class.

Use of git features

Of the teacher survey respondents who use git in their courses, nearly all of them use commits, push / pull, and branches

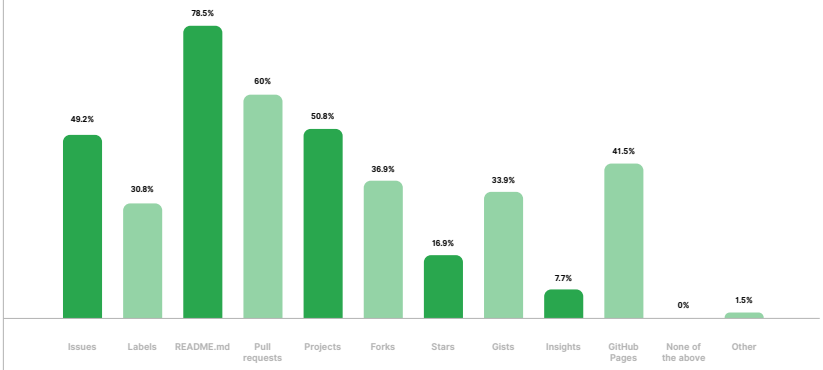


When it comes to GitHub, teacher adoption is lower, though still common. Over half (62 percent) of the teachers who responded to the survey use GitHub in their courses and of those who don't, most would like to. Of the teachers that do use GitHub, a

majority (63.6 percent) are using GitHub Classroom—a tool that helps educators with the workflow of teaching with GitHub—to implement GitHub in their courses.

Use of GitHub features

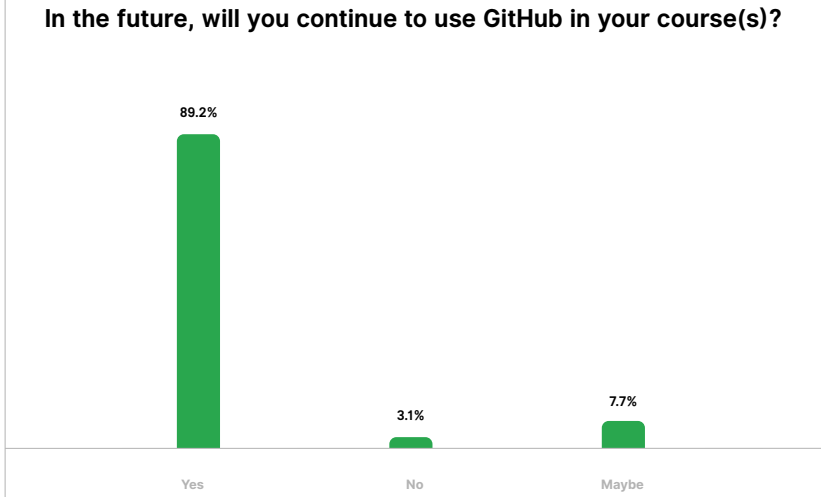
Of the teacher survey respondents who use GitHub in their courses, over half of them use README.md, pull requests, and projects



There's good news for teachers that plan to implement GitHub in the future: A little over half (50.8 percent) of teachers said it was easy to implement GitHub in the classroom, and nearly all of the teachers who use it now plan to continue to use it in the future.

Future use of GitHub by current users

Nearly all teacher survey respondents who currently use GitHub in their course(s) felt they will continue to use GitHub in their course(s)



This aligns with teachers saying they would recommend GitHub to others. Across all teachers who responded to our survey, GitHub got a high Net Promoter Score of 60. But with teachers who used GitHub in their courses or used GitHub Classroom, scores went even higher, to 70 and 76, respectively.

Net promoter score: Teacher survey

How likely is it that you would recommend GitHub to a friend or colleague?

	Overall	Current users	Implements GitHub in class	Uses GitHub Classroom
NPS	60	67	70	76
Detractors	7%	3%	5%	5%
Passives	27%	27%	21%	14%
Promoters	66%	70%	74%	81%

Taken together, this survey data shows that teachers can have a big impact on the skills students acquire in their coursework, but it also shows that there's more to using GitHub in the classroom than a demo or an assignment that marks the "version control" checkbox. To see how teachers are embedding GitHub into their instruction, we recommend revisiting our case study "How GitHub Classroom and live feedback improved students' grades".

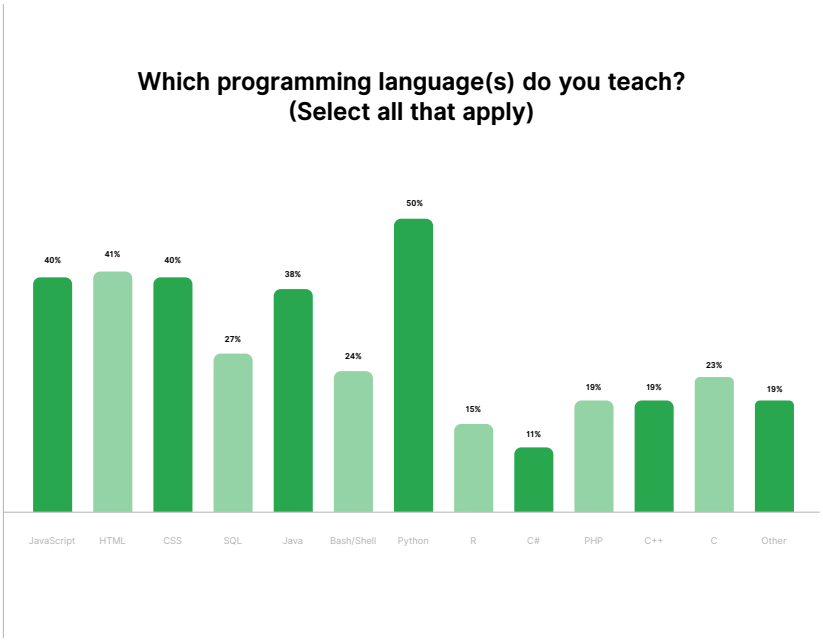
If you're a teacher and you want to bring GitHub into your classroom, then get started by visiting https://classroom.github.com.

Teachers are trendsetters in tech

Teachers aren't passively chasing trends in technology. They're integral to the trendsetting process, in part through the tools they select and in part by preparing their pupils to learn on their own.

To highlight one set of skills in particular, take a look at programming language use. Teachers aren't imparting knowledge exclusive to the classroom. They're teaching the languages that students go on to use in the workplace, such as Python, HTML, CSS, and JavaScript.

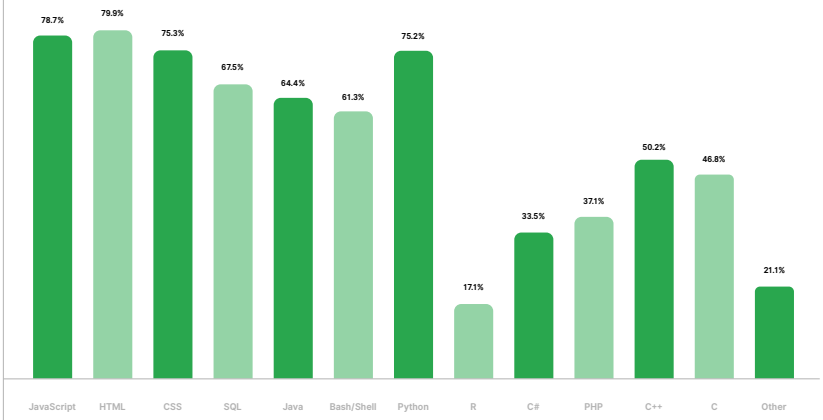
Programming languages taught



Programming languages

HTML, JavaScript, CSS, and Python are among the most commonly used programming languages by respondents who redeemed student coupons in 2014-2016

Which of the following programming language(s) have you used? (Select all that apply)

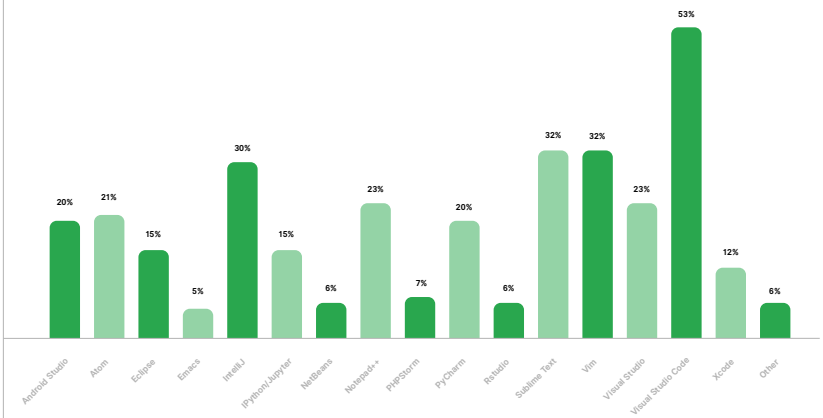


When we asked teachers specifically about what IDEs they used in their classrooms, they had similar proportional responses. Teachers, almost as much as student respondents, have incorporated IDEs like Visual Studio Code into their toolbox.

Integrated development environments

Visual Studio Code, Sublime Text, Vim, and IntelliJ are among the most commonly used IDEs by respondents who redeemed student coupons in 2014-2016

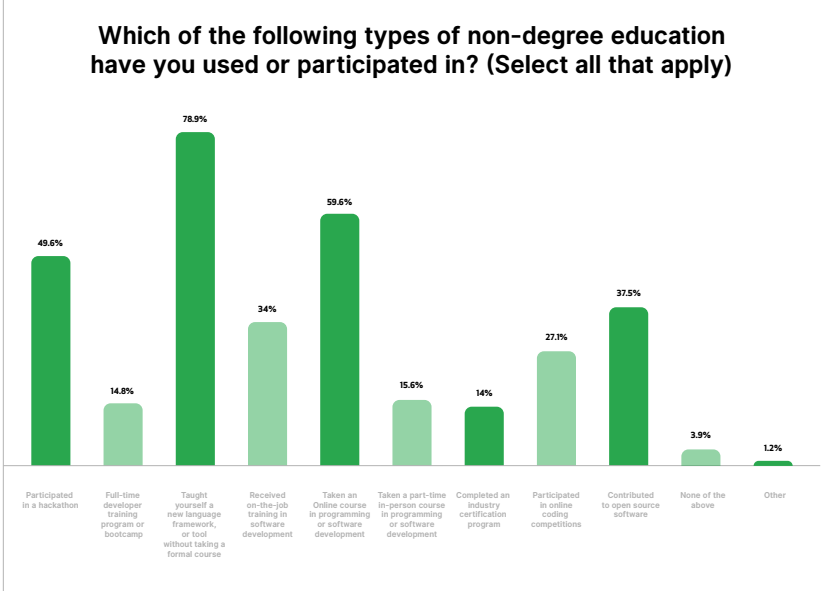
Which of the following integrated development environments (IDEs) do you typically use? (Select all that apply)



This trendsetting process is not a one-way street, though. Students are learning on their own, through hackathons, bootcamps, and open source contributions.

Participation in non-degree education

Most respondents who redeemed student coupons in 2014-2016 have taught themselves a new language, framework, or tool without taking a formal course



Teachers may be well-served by encouraging their students to seek out hackathons, for example (which we covered in more detail in “Survey results: Better computer science learning through hackathons”). Major League Hacking conducted a survey that showed students learn skills in hackathons that they bring back to their coursework.

If you’re a teacher and you want support for events or training to master Git and GitHub, there’s no better route than to become a GitHub Campus Advisor.

Conclusion

The Classroom Report for 2019 shows that GitHub is in the classroom and the workplace, now and in the future. Teachers are already using GitHub in the classroom, and more plan to teach it in future semesters. For students, if GitHub isn't an essential tool today, it will be as they continue their studies and enter the workplace.

GitHub

Learn more about GitHub Education programs:

GitHub Education
https://education.github.com

GitHub Campus Advisors
https://education.github.com/teachers/advisors

GitHub Campus Experts
https://githubcampus.expert

GitHub Campus Program
https://education.github.com/schools

GitHub Classroom
https://classroom.github.com

GitHub Student Developer Pack
https://education.github.com/pack

Contact us at
education@github.com