The Emergence of GitHub as a Collaborative Platform for Education

Alexey Zagalsky, Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, Weiliang Wang
University of Victoria
Victoria, BC, Canada
{alexeyza, noelf, mstorey, evianz, weiliang}@uvic.ca

ABSTRACT
The software development community has embraced GitHub as an essential platform for managing their software projects. GitHub has created efficiencies and helped improve the way software professionals work. It not only provides a traceable project repository, but it acts as a social meeting place for interested parties, supporting communities of practice. Recently, educators have seen the potential in GitHub's collaborative features for managing and improving—perhaps even transforming—the learning experience.

In this study, we examine how GitHub is emerging as a collaborative platform for education. We aim to understand how environments such as GitHub—environments that provide social and collaborative features in conjunction with distributed version control—may improve (or possibly hinder) the educational experience for students and teachers. We conduct a qualitative study focusing on how GitHub is being used in education, and the motivations, benefits and challenges it brings.

Author Keywords
CSCW; CSCL; Learning; Education; GitHub; Social Media; Qualitative Methodology; Distributed Version Control

ACM Classification Keywords
K.3.1 Collaborative Learning: Computer Uses in Education

INTRODUCTION

GitHub is a popular Web-based social code sharing service that utilizes the Git distributed version control system. It has become an essential tool in technology areas that require collaboration, such as software development and technical writing [33]. It is also seeing widespread adoption in other areas[1], transforming how people collaborate over a shared repository [2]. One of GitHub's main strengths is in the awareness and transparency features it provides to team, project and community members [5]. These features positively influence how people contribute to projects [36].

In this paper we examine how GitHub is starting to emerge as a collaborative platform for education. We aim to understand how environments such as GitHub—environments that provide social and collaborative features in conjunction with distributed version control[2,3]—may improve (or possibly hinder) the educational experience for students and teachers.

Only a few years after GitHub's 2007 release, well-known computer science educator Greg Wilson suggested[4] that GitHub could be used for learning materials despite some limitations:

> *Would it be possible to create a "GitHub for education?" Right now, I think the answer is "no", because today's learning content formats make* ***merging hard****. Whatever a "GitHub for education" would look like, it would not be yet another repository of open learning materials. There are lots of those already, but almost all their content is write-once-and-upload, ... rather than* ***sharing course content in a reusable, remixable way****.*

In 2012, he further elaborated[5]:

> *"GitHub for Education" isn't necessarily, "Let's put educational materials in GitHub", but rather, "Let's* ***facilitate a culture of spontaneous-but-structured collaboration*** *and improvement."*

He recognized that the majority of learning management systems introduced friction for instructors trying to reuse and share course materials—the type of problem that the software development community solved through the use of tools such as GitHub [5]. Moreover, his comments echo the **participatory culture** that, according to Jenkins [17], results from the use of social media in education.

In an effort to promote GitHub to higher education, GitHub launched the GitHub Education Website[6] in 2014. In re-

[1] `http://www.wired.com/2013/09/github-for-anything/`

CSCW '15, March 14–18 2015, Vancouver, BC, Canada
Copyright 2015 ACM 978-1-4503-2922-4/15/03$15.00
http://dx.doi.org/10.1145/2675133.2675284

[2] `https://bitbucket.org/`
[3] `https://www.gitlab.com/`
[4] `http://software-carpentry.org/blog/2011/12/fork-merge-and-share.html`
[5] `http://software-carpentry.org/blog/2012/04/github-for-education.html`
[6] `https://education.github.com/`

sponse to this and other *ad hoc* opinions on the benefits of using GitHub for education, we conducted a first study to examine whether and how GitHub, with its powerful collaboration, social and awareness features, can be used for educational purposes. Specifically, we aimed to answer the following research questions:

RQ1: **How does GitHub support learning and teaching?**

RQ2: **What are the motivations for and benefits of using GitHub for education?**

RQ3: **What challenges are related to the use of GitHub for education?**

We conducted interviews with early adopters of GitHub for educational purposes and followed-up with a survey to validate our findings. Our findings indicate that GitHub not only supports the main learning features present in traditional Learning Management Systems, but it also provides additional features that educators and students can benefit from: revealing transparency of activity, encouraging participation and collaboration, and supporting reuse and sharing of course materials and knowledge. However, it also brings challenges, such as a difficult learning curve for students. We conclude with a discussion of our findings and recommendations for GitHub users.

BACKGROUND

Before discussing GitHub and how its unique features can support education, we present a brief overview on technology in education and the strong connection between education and Computer Supported Cooperative Work (CSCW).

CSCW and Education

In the mid-1990s, online learning was supported by the Internet, email and forums, and focused on interactions between the student and the computer. Currently, online learning focuses on computer-mediated communications and the formation of communities of practice [26], and uses technologies such as cloud computing [34], social media (e.g., Twitter [15, 31]), mobile devices [12] and Massive Open Online Courses (MOOCs) [28]. Technology has had a major influence on education and it will continue to do so (e.g., Khan Academy [35]). However, questioning if and how various technologies should be used to improve learning is non-trivial.

As described in the book "Technology in Education—Looking Toward 2020" [27], participants of the "2020 Panel" tried to envision the future of education and enrich people's understanding of technology's potential role in education, as well as opportunities, drawbacks and challenges. William Bossert, one of the panelists, proposed that technology should be used to encourage intellectual exchanges among students in the classroom. *"Appropriately designed computer programs would make possible the solution of complex problems by a group of students, while helping 'level the abilities' within a group so that less able students might take on tasks that would be viewed by others as meaningfully furthering the group activity."* Another panelist, Jim Minstrell, envisioned the classroom of the future as *"full with electronic aids for teachers, helping with organizational and management tasks such as: keeping track of appointments and deadlines."*

Moreover, a branch of CSCW—Computer Supported Collaborative Learning (CSCL)—describes the benefits of collaborative learning using technology. Creativity in a collaborative context, for example, can be aided by activity awareness, whereby each individual is aware of other collaborators' work and is cognizant of other people's plans and understandings [11]. Indeed, Web-based collaborative learning systems give learners more opportunities to participate, which may benefit both individual and team performance [23]. Our findings suggest that GitHub is one such system, as the use of its collaboration, social and awareness features provides benefits beyond traditional systems. This transforms the ecosystem of teachers, students and content into an open collaborative ecosystem (see Fig. 1).

Learning Management Systems

Typically, a learning management system (LMS) or course management system (CMS) (e.g., Blackboard, Moodle, Sakai) provides educators with course management features that allow grade tracking, file management, assignment hosting and real-time chat. Many of these are listed in Kumar *et al.*'s comparative study of different virtual learning management systems [20]. Malikowski *et al.* [25] developed a model that dissects the quality of LMS tools into five categories: (1) transmitting course content, (2) evaluating students, (3) evaluating courses and instructors, (4) creating class discussions, and (5) creating computer-based instruction. Their study shows that the most prominent use of an LMS is to transmit information to students, whereas the categories of creating class discussions and evaluating students receive moderate and low-to-moderate use, respectively.

In response to the popularity of Web 2.0 technologies, LMSs evolved to include features such as blogs and wikis [9]. However, learning systems are typically geared towards administrative purposes rather than fully supporting self-governed or collaborative learning activities [6], leading to interactive features being underutilized and students being less engaged [1]. We argue that GitHub can support much of what traditional learning systems do, as well as go beyond them by supporting collaborative activities.

What is GitHub?

GitHub offers several unique features to facilitate user collaboration. Its most important feature is the Pull Request (PR) mechanism which is a way to initiate discussion with other users and share or comment on the various artifacts in a project (typically changes to the project's content). The discussion may include code that is visible to everyone and it shows the exact changes that would be merged if the PR were accepted. A PR may involve other content (e.g., screenshots) to provide a background for the discussion, or include changes to other resources in the project. When a user wishes to contribute to someone else's project, they can *clone*[7] the project to create a full copy of the project in their

[7] `https://help.github.com/articles/duplicating-a-repository`

local environment, but where committed changes will still affect the original project. This is called a Shared Repository Model[8]: contributors can either commit changes directly into the shared repository or use PRs to start code reviews and conversations about proposed changes before the changes are merged into the master branch. Alternatively, a user can *fork*[9] the entire project to create a parallel project where committed changes do not directly affect the original project. This is called a Fork & Pull Model: PRs provide a way to notify the original project maintainers about the changes you would like them to consider.

Users can not only follow other users or projects of interest, but they can also broadcast their activities to their followers. Furthermore, users can discover new projects by using the *explore* feature, or share snippets using the *Gist* feature. GitHub also supports awareness by broadcasting updates to the user's news feed. The combination of these features facilitates *"a culture of spontaneous-but-structured collaboration."*[10] As an example, Dabbish *et al.* [5] describe GitHub as beneficial to the collaboration of a community because of transparency, where individual contributors can infer a collaborator's technical goals or be aware of what other users are paying close attention to.

GitHub is not just a prime example of CSCW, but it is also a good data source to study how people collaborate[11]. Wu *et al.* [38] analyzed rich GitHub timeline data and found that most of the *follow* activities on GitHub are not due to developers' collaborations on GitHub, but rather their interactions outside GitHub, such as through Twitter, Hacker News and Stack Overflow. Majumder *et al*. [24] conducted an investigation on how teams are formed in a social network based on millions of software repositories spanning a period of four years and hundreds of thousands of developers on GitHub. Tsay *et al*. [36] performed a quantitative analysis of project success of over 5,000 open source software projects hosted on GitHub. They used two measures for project success—**developer attention** and **work contribution**—and found that projects with a high level of developer multitasking tend to receive less developer attention, but greater work contribution.

Why GitHub for Education?

Software developers, sometimes described as the "prototype of the future knowledge worker"[12], are often the first to adopt new tools and new techniques. We predict that future knowledge workers will have to deal with more complex models for managing content than has been done in the past and how GitHub is being used for education is one example of that.

Up until recently, GitHub has focused on code and project management for software development; it is now being extended to other domains that involve collaborative work, such as education [14]. However, in this different context, teaching techniques may require adjustment while some of GitHub's features may require repurposing. Jim Baker, a senior developer and University of Colorado Computer Science lecturer, shared his experiences with GitHub:

> *"We had a great experience using GitHub to support a collaborative workflow for the 70+ students in each of the 2 semesters of my CS course."*

When asked for more details, he elaborated:

> *"Pull requests (PR) are the heart of the GitHub workflow, and we took advantage of PRs, including task lists so that students could report on their work in progress and get over initial humps. Any merged PR got extra credit(!). Because the course had been improved in some way—this seemed like an interesting standard for giving out extra credit. Consequently, we mostly didn't merge PRs for labs, except for bug fixes, but we were always on the lookout for* ***better solutions than ours****. PRs were also merged for extra credit, such as* ***corrections of my course notes****. Next fall we expect to have* ***autograding*** *implemented as a form of continuous integration, by running against the PRs through postcommit hooks."*

Several other educators have also introduced GitHub into their classrooms and shared their experiences. In 2010, Luis Felipe Borjas[13] posted about the use of GitHub organizations[14]—a way to simplify the management of group-owned repositories—to manage class projects. He also applied GitHub to exams: instead of waiting until the exam deadline to upload the exams, students could just *push*[15] as many times they wished and the last *push* they made before the deadline was going to be considered their final submission. He also strongly suggested teachers create exams or homework assignments that could translate into private *git repos* for students to push to.

In 2011, David Humphrey blogged[16] that he asked his students to use Git/GitHub and highly recommended other instructors use it in the classroom. He claimed that although it was a little painful to learn Git/GitHub at the beginning, the payoff would be huge. *"One of the great things about Git in an educational setting is that you don't need to rely on institutional IT, which, in my experience, is never agile enough to help you with revision control. You can put repos on laptops, use USB keys, use DropBox, use GitHub, etc. You don't have to wait for someone to set up a server and make you accounts, don't have to deal with permissions, or any other nonsense that comes with centralized revision control systems."*

Version control systems have been used in the classroom as a way of managing students and their work. Reid & Wilson [30] introduced the Concurrent Versions System (CVS)

[8] https://guides.github.com/introduction/flow/

[9] https://help.github.com/articles/fork-a-repo

[10] http://software-carpentry.org/blog/2012/04/github-for-education.html

[11] http://www.gousios.gr/blog/The-triumph-of-online-collaboration

[12] http://allankelly.blogspot.ca/2014/04/the-prototype-of-future-knowledge.html

[13] http://lfborjas.com/2010/10/30/git-classroom-exams.html

[14] https://github.com/blog/674-introducing-organizations

[15] https://help.github.com/articles/pushing-to-a-remote

[16] http://vocamus.net/dave/?p=1358

to a second-year computer science course. This provided the instructors with a simple way to manage student assignments, made it easier for students to work in pairs or groups, and gave the instructors a history of student work. Clifton, Kaczmarczyk & Mrozek [4] used Subversion, another version control system, to collaboratively develop and run introductory computer science courses. The ease of managing courses using Subversion allowed the instructors to free up time from administrative demands, allowing them to spend more time focusing on pedagogical issues. In 2013, Griffin & Seals used GitHub in the classroom as a version control tool, leveraging the *Branch* and *Merge* features [14]. When students worked on programming assignments, it was easy to *merge* back into the original project if their version worked, or abandon a branch without destroying the original project.

Apart from these testimonies of using *"raw GitHub"* and other similar, developer-focused tools in the classroom, there are also education targeted platforms based on the GitHub model. Created in 2013, Coursefork[17] is described as *"GitHub for course creation."*[18] It is a platform for open-sourcing and collaborating on educational material, where educators can upload course materials and allow others to create copies of courses and modify or share them.

These examples show that GitHub is generating interest as a powerful platform for education. Our research provides first insights into how GitHub can benefit education.

METHODOLOGY

To discover if and how GitHub is being used in education, our study used exploratory research methods [10, 32] and consisted of three phases of data collection (online methods [37], interviews, and a validation survey). Our methodology consisted of data collection and iterative phases of data analysis. In this section, we discuss our research questions, study design, study participants, data collection methods and data analysis approach.

Research Questions

This study strives to answer the following research questions:

How does GitHub support learning and teaching? We investigate how GitHub is used and for what purposes within the education domain. Our findings indicate that even though GitHub use in education mirrors the way traditional learning systems are used, the implications differ substantially.

What are the motivations and benefits of using GitHub for education? Based on testimonies from our study participants, we explore the motivations for GitHub use in education and the possible benefits it might bring to support learning. We also look at the specific features of GitHub that are being used to support learning.

What challenges are related to the use of GitHub for education? And finally, we examine the challenges educators and their students face when using GitHub to support learning and teaching. We provide specific examples based on interviews with educators, and later synthesize recommendations for educators wishing to use GitHub, as well as recommendations to the GitHub design team.

Study Design

In the first phase of our study, we used online research methods [37] and searched for resources (such as blog posts) that described the personal experiences of educators using GitHub to support learning or teaching. The results from this phase are presented in the Background section.

Through this exploratory phase, we found creative and successful examples of GitHub supporting teaching and learning in the classroom. We also found discussions on the challenges and difficulties of using GitHub. This allowed us to refine our research questions and guided us in the following phases of the study, such as shaping the questions to ask during interviews.

In the second phase of our study, we interviewed 15 participants, including one of the blog authors from the first phase. In this phase, we were able to thoroughly investigate the usefulness and potential of GitHub in education. Through iterative analysis of the data collected, several themes emerged around the motivations for and challenges of using GitHub to support learning.

These themes informed the third phase of our research: a follow-up survey sent to interviewees from the previous phase and to other educators using GitHub for education. The goal of this survey was to receive interviewee feedback on our interpretations, but also to gain additional perspectives from other educators that use GitHub.

Data Collection

For the interviews, we emailed invitations to lecturers and professors that use or have used GitHub to support teaching or learning. Potential participants were recruited in several ways: by contacting blog authors who shared their experiences of using GitHub in the classroom; by posting an invitation on Twitter[19]; and through snowball sampling, where interviewees could suggest other colleagues.

The interviews lasted 20-60 minutes and were conducted face to face or with Skype. Audio was recorded and the interviewer took notes. The interviews were semi-structured based on 18 guiding questions (our interview form is available online[20]), and the interviewer could *dig deeper* with additional questions as deemed appropriate. This supported the exploratory nature of our study and allowed us to examine interesting and unexpected insights.

We also interviewed John Britton, an education liaison from GitHub (this interview form is also available online[21]). This

[17] http://coursefork.org/

[18] http://opensource.com/education/13/9/coursefork-education-tool

[19] https://twitter.com/alexeyzagalsky/status/465914075348619264

[20] https://drive.google.com/file/d/0B273SDwCaBm_Z3V1X1BtakpKVlk/edit?usp=sharing

[21] https://drive.google.com/file/d/0B273SDwCaBm_Z2toY0hBUWVjZ3c/edit?usp=sharing

interview provided us with GitHub's corporate take on the use of its tool in education.

Participants

Our study targeted lecturers and professors in higher education who use or have used GitHub to support teaching and learning. As our study aimed to investigate diverse populations as well as GitHub's usefulness in non-technical courses, we wanted to hear from lecturers and professors across domains.

Interview Participants

Table 1 shows some details about the courses each of the interviewees taught using GitHub. For each course taught, we list a summary of the students' GitHub knowledge, the type of course, and the number of students enrolled.

Table 1. Information on courses taught by interview participants while using GitHub.

P#	Course Type	Knew GitHub	Course Size(s)
1	CS	Yes	55
2	CS	Yes	40
3	CS	Yes	85, 130
4	Humanities	No	11, 40
5	Humanities	No	17
6	CS	Yes	60, 100
7	CS	Yes	50-237
8	CS	Varies	60
9	Sciences	No	235
10	CS	Varies	450
11	CS	Varies	20, 40
12	Statistics	No	40
13	CS	No	20
14	CS	Yes	8
15	CS	No	30-32

The participants represent a broad list of universities: University of Victoria, University of British Colombia, McGill University, University of California at Berkeley, University of California at Davis, Columbia University, The City University of New York, Harvard University, The University of Texas at Austin, Federal University of Pernambuco, and Delft University of Technology. Furthermore, one of the participants was from a company that teaches code to entrepreneurs and CEOs in Paris.

Follow-up Survey Respondents

We sent a follow-up survey to the interviewees to get their feedback on our interpretation of the interview findings. We also publicly broadcasted the survey using social media[22] and used snowball sampling to garner responses from additional educators we had not interviewed. 8 of the interview participants and an additional 7 new respondents completed the survey, for a total of 15 responses.

Data Analysis

Our analysis of the interview data followed qualitative data analysis guidelines [21, 32] and included the following stages: (1) transcription of the data recorded; (2) organization of the data into easily retrievable sections; (3) familiarization by reading and re-reading the data, making memos and summaries; (4) reading the data and labeling segments, i.e., coding; and (5) identifying themes or emergent concepts through discussions among the researchers and engaging in re-coding to develop more well-defined categories. After refining and merging some of the themes, a list was compiled. This process was performed iteratively for each category.

[22] `https://twitter.com/alexeyzagalsky/status/471053256718692352`

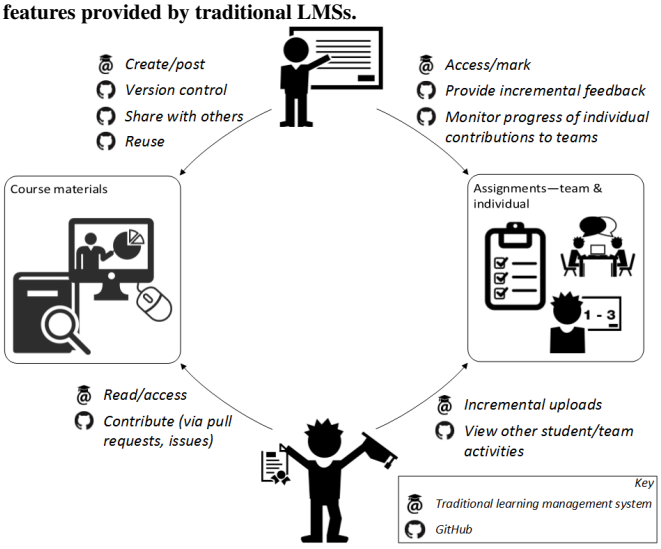
FINDINGS

In this section, we address our research questions and present the themes that emerged from the interviews. To illustrate the different aspects of each theme, we provide selected quotes from the interviews, where each participant is identified by an anonymized identifier (P#).

RQ1: How does GitHub support learning and teaching?

There are two ways that our interviewees utilized GitHub in the classroom: as a submission platform, and as a way to host course content. These two basic uses mirror how teachers use typical learning management systems (e.g., Moodle and Sakai) [25]. However, the implications of how they use these features differ substantially. Figure 1 gives a high-level overview of the interactions that are possible between teachers, students and content, and shows the additional interactions that the use of GitHub natively supports over traditional learning environments. For example, as seen in the bottom-left quadrant of the figure, while traditional LMSs support students reading and accessing course material, GitHub also allows them to easily contribute to the course material.

Figure 1. The additional features GitHub provides in addition to the features provided by traditional LMSs.



GitHub as a Submission Platform

Many of the interviewees used GitHub as a place for students to host their work and submit their course assignments and projects [P1, P2, P3, P4, P5, P6, P7, P11, P12, P13, P14, P15]. The benefits of using it as a submission platform are further discussed in the findings of RQ2.

Using GitHub as a submission platform was accomplished in one of two ways[23]. Some interviewees set up a base repository for the class and had each student fork it—everyone who forked the base repository could see all other forked repositories as well. This allowed students to cross-reference different solutions and encouraged student collaboration (e.g., by using PRs) and peer learning. *"When you do a pull, you can see what the others in the seminar are doing. For example, a student wrote a python script, and others want to use it, so they can just grab it and use it."*[P4]

Other interviewees set up private repositories for each student, and individual students could only see other repositories when explicitly given access. For P5's course, *"Students have separate repositories, and it's private so that people cannot see what they are working on."* With this option, repositories and permissions are manually set up, which may be difficult in courses with a large number of students.

Hosting Course Material

Many interviewees used GitHub to host and deliver course materials [P4, P5, P8, P9, P10, P11, P12, P13, P15], including syllabi, slides, notes, reading materials, exercises and homework. It was even used to host the actual course Website, making GitHub *"a bulletin board."*[P10] The basic mechanism to host course Websites is by using GitHub markdown files, or for more traditional Websites, by using GitHub Pages[24] which hosts the material directly from one's repository.

With course materials hosted on GitHub, students and other educators were able to suggest course improvements [P1, P10, P11, P12] by submitting an issue or a PR. *"Students would actually find corrections and things and they would send me Pull Requests... But there was some sense of a community that was at least, certainly invited to collaborate on the course material itself."*[P12] As such, interviewees were able to facilitate student contribution to the course material itself using GitHub, a use case and benefit that GitHub offers beyond many traditional systems.

RQ2: What are the motivations and benefits of using GitHub for education?

We explored why educators chose to use GitHub and how they (and their students) benefited from its use. While GitHub is still an emerging tool for education and many benefits have not yet been fully realized, we extracted several themes from our interviews.

Transparency of Activity

Using GitHub as a submission platform made it easy for our interviewees to monitor student progress, activity and participation. GitHub has numerous features that support transparency of student activities. For example, GitHub allows users to see the **history** of activities. As P1 mentioned, *"You really see the full history of how the document comes into being, including all the discussions, the former versions. And the groups can look at each others' documents and see them emerge. I can see how they are creating it, so I can monitor who's active, working in certain teams, which is also handy, practical."*

GitHub also provides a **graph view** that visualizes a summary of project activities. This allowed teachers to easily gain a high-level view of student activity. *"You can see who is doing what, you can see how people are sharing work because you have the commit history."*[P4] This same feature supported teachers in identifying outlier students that were not participating as expected. *"I would go into their team repository, and...they have some graphs in there that show you what's the activity level and who's most active. I would use that to get a sense for what was happening because inevitably, you would have one of the students...that's not doing anything and you want to get some hard data on exactly what he isn't doing."*[P6]

Furthermore, GitHub has **news feeds** that teachers used to keep up to date with activity. By watching the news feeds, they could catch problems early with how frequently students were participating in the course. *"Personally, what I did is that I subscribed to all [the students'] repositories and followed the feed of those repositories."*[P2]

However, GitHub allowed people to do more than just passively observe progress. Students **opened issues** and **tagged** them with the teacher's name to get the teacher's attention. *"You could watch the commits go by. They can ask for help by opening issues and tagging us, so that worked quite beautifully for monitoring and in some cases actively providing advice on group project work."*[P12] This kind of support encouraged a style of communication between the teacher and student that is not available in traditional learning management systems.

Encourage Participation

The transparency features discussed above allowed interviewees to encourage student course participation and contribution to the hosted course material. With traditional environments, students would need to contact the instructor directly to suggest changes to the materials. Using GitHub, a student could suggest changes by submitting an issue connected with the material. Or, they could make changes themselves and then submit a pull request, which the teacher, if they agreed with the changes, could simply accept. The history of these actions helped the teacher keep track of who participated, and by its visibility, also encouraged students as their actions were logged and possibly used to improve their grades.

P1 spoke about how these features helped to persist the effort that students put forth: *"Yes, [using GitHub encouraged student participation] because everything you do is also recorded. It stays there forever, it's persistent. So your comments are persistent, your activities are persistent. So you work, you get rewarded, and if you don't work it's visible... in GitHub it's a little more transparent."*

Some interviewees felt that using GitHub helped them encourage participation, even in indirect ways: *"I will use the logs [that students put on GitHub] as materials for discus-*

[23] `https://education.github.com/guide#4-set-up-the-repositories`

[24] `https://pages.github.com/`

sion in class. That can encourage participation because you know you are committing something that can be discussion material."[P5]

Although few mandated the use of GitHub's issue tracker for their classes and students' group projects, some found novel uses for this feature. P12 and P14 had students use it when they needed help, and otherwise used it as a simple communication method between students and markers. P1's students participated by discussing an issue they had with a deadline: *"the students didn't like [a deadline], so they opened an issue on GitHub: 'Can we move it?'... So everyone responded to that issue on GitHub, and then at the end I said, 'based on the discussion, it is going to be on Sunday."'* These features supported student/teacher discussions—communication was recorded and tied directly to the relevant course content.

Reuse and Sharing of Course Materials and Knowledge

Another important advantage GitHub provided our interviewees was the ease with which materials and knowledge could be shared and reused, by both students and course instructors.

Some of our interviewees' students shared work and material with each other. In P4's course, a student's python script was visible to others, and therefore, easily shared. The ability to make content either public, private amongst collaborators, or completely private, gave our interviewees' students an easy way to share with others or contribute to their work. For example, P1 noted that this visibility had the useful side effect that his students gave each other **feedback** on their pull requests and reports.

Moreover, our interviewees used GitHub to share courses and their associated materials with other educators. Although this is also a feature in many learning environments, they lack a way to visualize how a reused course has changed from the original one. Seeing when courses are **forked** and how they differ from one another is a key feature in GitHub. *"We were developing all the content from scratch... the various teaching staff had to collaborate and develop the material, and so GitHub was a nice fit for that... We had two repositories: we had a private one and a public one, and so we could sort of have private conversations among the staff to refine the content and then it was pretty easy to just migrate it over once it was ready."*[P10]

GitHub's **access control** features allowed the instructors to **version** their material and choose who to share it with. *"Some of [the material] is shared with the world, and some of it is shared with other instructors of other universities. So for example, when we create new exam questions and stuff like that, we try to keep those [repositories] private for a little while so other instructors can use them. But we're versioning all of it."*[P7]

Interestingly, unregistered students and other GitHub users would also visit these public repositories: *"We kept everything public from the beginning, so it ended up getting a lot of attention outside of the course. I think the repository has something like 200 stars on GitHub right now and as far as I can tell most of those stars are from people who didn't take the course."*[P10]

Industry Relevance

As working with Git and GitHub are relevant skills for industry, this theme emerged as both a motivation to use GitHub in courses and a benefit of using the tool. This relevance to industry meant that some of our interviewees and their students had prior GitHub experience, and as a result, were motivated to use it in class. For example, P12 had already been using GitHub regularly and explained why they chose to use GitHub in their courses: *"... partially to unify what I'm doing in the research side of my life with what I do with the teaching side. Like once you've taken the trouble to learn [GitHub] and get it all set up, you start to see lots of areas of your life where this workflow would actually be very useful."*

Meanwhile, we saw some instances where students who had prior experience using GitHub encouraged their course instructors to use it for class. As P1 told us: *"So actually, some students came to me, and they said - you should watch this video on how GitHub uses GitHub to build GitHub."*

Learning to use Git and GitHub in a course is a significant benefit in certain fields, particularly in computer science and software engineering[25]. In fact, P12 felt that learning to use GitHub was required to progress within their field: *"To prepare this cohort of graduate students for computational work, they should know how to use these tools. This is very much what people do these days in [statistics], and so I actually consider it a completely valid pedagogical goal in it of itself."*

By using GitHub for course work, students further benefited by having an end product that could easily be shared with prospective employers. *"you can use it sort of as an online resume... GitHub allowed you the opportunity to convert [your class work] to a public project so that employers for example could see what code you've been writing."*[P6]

Ease of Use

GitHub's administrative functions are relatively simple and easy to work with: getting students set up for the class, setting privacy options and creating repositories. *"I don't think it saved time at the beginning... because I was starting an entirely new approach to doing this. But once it was up and running, the next semester would have been quite simple to administer because I knew the process."*[P6] Although GitHub has a definite learning curve, it can be mitigated by sharing practices among practitioners. *"I worked with [the next instructor of the course] a bit on explaining how it worked and what we did with it, so she found it pretty easy to get started too."*[P6]

Despite the learning curve and some technical difficulties, several interviewees mentioned that GitHub was relatively easy to use and to administer, both on its own and compared to existing university systems. *"For me it was the ease of updating the class schedule and course notes. The class schedule is something that is extremely painful to update on the university Website... Each course is a little window and you have to click arrows to move them up and down to reorder them, or something really horrible like that. And [with GitHub] it*

[25] `http://techcrunch.com/2014/07/23/modernizing-computer-science-education`

was very easy to just go into a markdown, add a link, and hit push."[P9]

Free Academic Licenses

Generally, GitHub allows all users to create free public repositories. However, an additional benefit mentioned in the interviews [P3, P6, P7] is that GitHub provides free academic licenses. Students and educators can apply for a free micro plan that allows private repositories. Educators can also apply for a free organization account[26] that facilitates team management and administration. The implications of these free public accounts go beyond benefiting face-to-face instruction, allowing teachers to *scale-up* traditional education and create online courses that will benefit part-time and remote students.

Shared Space and Course Versioning

By hosting course materials on GitHub, students could easily share course notes, references or other material for the same class. From the aspect of teaching, educators could easily share [P13] or duplicate course resources. *"The simplest thing I've been using [GitHub for], especially from an instructor's view, is to duplicate the course Website for different semesters."*[P4] Therefore, when teaching the same course again, preparing the course materials was simple: participants used the same repository or forked it. *"If I improve the material in some way, I'd just keep it there...And if someone wants an older version of the repository, they just need to get it. However, if I did want to create a separate course with just part of the material of the original course, I would [fork the repository]."*[P11]

RQ3: What challenges are related to the use of GitHub for education?

We have grouped the challenges our analysis revealed into five themes: shared knowledge base of suggested practices, barriers to entry, support for additional formats, external restrictions, and large scale management.

Shared Knowledge Base of Suggested Practices

Interviewees showed a need for *a shared knowledge base* of suggested practices on how to use GitHub for educational purposes. P6 went into great detail about their struggles in this area: *"It's a different use in a team at some startup or what have you. And it's not clear how exactly you should, for example, enforce a methodology for the students for working. Should you use Pull Requests? When should you file issues? Divide up the repository, should you each have an individual repository as well and then fork the main project? There's a lot of good evidence and good data for how you should work in Git for your software development project, for commercial projects. For educational projects, they're quite different because they're short in duration, there's 4 or 5 people working on it that aren't very experienced with software development."*

Educators wished for easily accessible *how-to guides* or shared experiences by others on how they've used GitHub. *"Maybe some documentation on best practices or some kind of shared knowledge base that would say here's what so-and-so at UC Irvine is doing with GitHub."*[P6] Interestingly, the GitHub education Website contains a guide that could meet some basic needs, but our interviewees did not know of this relatively new feature.

Barriers to Entry

At its core, GitHub is a **Git-based** system, and while it provides a simple Web interface, tasks that involve collaboration require an understanding of Git and its command-line arguments. In particular, dealing with **merges and conflicts** can be challenging: *"To get the most out of GitHub, you need to understand Git. Even if you just use it to edit documents together, you will get conflicts once in a while. And if you want to use the review mechanism, then you want/should use the Pull Requests. That requires some pretty deep knowledge of Git still. If you use it the right way it is simple, but somehow with Git you end up with conflicts, and if you don't understand it, it's magic."*[P1]

Difficulties using Git are experienced by technical novices and software developers alike [29]. Our interviewees reinforced the need to improve **accessibility** for novice users or users with a limited technical background: *"Largely, the biggest challenge is to lower the learning curve, not on the high end, but the very low end for people who are new to it to have much gentler learning curve."*[P5]

Support for Additional Formats

GitHub's file and format support is another main challenge mentioned by the interviewees, specifically the lack of support for formats widely used in education, such as PDF and LaTeX. The ability to view or render these formats directly on GitHub, similar to Markdown rendering, would be very helpful. But more importantly, there is a need to support the powerful features Git has for text files: *diff* and review functionality, and the mechanism to add inline comments to a file without altering the original file. For example, when an instructor wishes to mark the students' assignments or reports, they are forced to download and comment on each file (e.g., using the comment feature in a PDF file) separately, which also alters the original file. Another option is to use the *issues* mechanism of GitHub to provide comments, however, this mechanism can't reference a specific place in the file and may be tedious if the instructor wants to add many small comments on a submission.

Poor support for slides was also mentioned: *"You can not easily view slides in GitHub, because the PDF is too large and you have to download it, it's a bit cumbersome... in a course you always have presentation material, so you want some sort of integration with slides here. They [GitHub] have Speaker Deck, so I want Speaker Deck integrated with GitHub."*[P1] Being able to *diff* slides is also important.

External Restrictions

Interviewees also mentioned external restrictions that limit or prevent them from using GitHub for education. External restrictions can be local restrictions (e.g., university policy) or global restrictions (e.g., regional publishing licenses). *"We*

[26] `https://help.github.com/articles/what-s-the-difference-between-user-and-organization-accounts`

have LMS systems that we use, and in a way it's very important to use them because they're authenticated by the university. So things like student's grades and stuff like that you don't (store on GitHub), because of the rules in the US. You cannot store it anywhere else."[P3] Another teacher also shared this concern: *"Ethically, I wish I could know where the server is, where I am pushing all my material."*[P4]

Knowing the server's location plays a significant role when educators decide whether to use GitHub. Not only from the **ethical aspects**, but also from **copyright aspects**. As P5 mentioned: *"Copyright is a big issue. For instance, we are working with a novel. In Canada, that novel is in public domain so it can be accessed online, but not in United States."*

Large-scale Management

Educators are challenged when managing courses with large numbers of students, teams, projects or issues. *"One thing I'm not so happy with, is the way group management [works]. So managing teams, different sizes, I have an organization for the students and the organization has 50 members and I need to create different groups, and different roles, and different access... I think that could be done... [in a] more convenient way."*[P1]

P6 further discusses the challenges with managing teams and repositories: *"The thing that I think was missing... was more management for the administrator: assigning people to teams, assigning teams to repositories, finer grained permission control, I think [that] is something that could have been useful."* However, the cause of these difficulties might not be an issue with GitHub, but perhaps the lack of *how-to guides* and suggested practices by others.

GitHub's Perspective

To gain insights into GitHub's perspective on the topic, we interviewed John Britton, an education liaison at GitHub and one of two engineers working full time to support educational uses of GitHub. GitHub's main goal in this regard is to make the tool easier to use, and to make sure users are aware of what resources are available to them to meet their needs and solve their challenges.

Regarding GitHub's benefits for education, John replied:

> *"We're working with students and teachers on using GitHub. It's essentially leveraging those tools in the classroom, so you can get a classroom experience that's more similar to what people in the industry are using, as software developers."*

However, there are challenges involved:

> *"When I first started, I definitely was targeting all forms of education. But I think that it's most applicable in computer science, software engineering, and technical fields. Maybe someday when the tools are easier to use and... require less technical knowledge upfront, it will be much more useful to non-technical fields, or to other types of educational stuff. But I think that Git as it stands requires a certain amount of basic knowledge of computing."*

He then elaborated:

> *"There's different groups. People who know Git and GitHub already and just want to use it in the classroom—they're totally on board with using Pull Requests right from the beginning. The group of people who's like, 'What's Git? What's GitHub?', they have a little bit harder time getting into the mindset of students submitting their code as a Pull Request. Rather than submitting an assignment with a zip file, you create a branch and make a Pull Request, and then grade the student on the Pull Request"*

John concluded by describing how GitHub plans to assist educators:

> *"I think we're going to do more of those [stories of existing use cases], along with technical documentation, or technical information on 'Here's what they're doing. Here's how you can do it too'. But as far as...we don't have a set, like, this is the way, right? There are multiple ways to use GitHub in the classroom, and it depends on what goal you're trying to achieve."*

The education use case is important for GitHub. While their focus remains on its use in software development, they provide dedicated support and resources for educators to take advantage of the tool.

Follow-up Survey

After conducting the interviews, we sent a follow-up survey to both interviewees and the public. We wanted to see if respondents agreed with our interpretation of the interview findings. However, the number of respondents (15) was too low to confidently validate our findings. Regardless, the survey supports many of our findings, with respondents mostly agreeing or strongly agreeing to many of the uses (as a submission platform and course material host) and benefits (simpler than current university systems, reuse and sharing of material, and facilitating collaboration). Interestingly, many respondents from this group of educators stated that they didn't struggle with the challenges that emerged in our interviews (e.g., technical difficulties limiting educator GitHub use, external restrictions, and managing large groups). However, while strongly disagreeing that technical difficulties limited their own use of GitHub, most respondents agreed that technical difficulties limited their student use of GitHub.

DISCUSSION

Our study uncovered how some educators use GitHub to support learning and teaching, while extending or even replacing traditional LMSs. However, the implications of our findings go beyond GitHub itself. The emergence of *"the GitHub way"* within education is transforming the traditional e-learning model (Figure 1) and will better support socio-collaborative learning environments [18] of the future.

Comparing GitHub to Traditional LMSs

GitHub was not designed as an LMS or a CMS, but our study shows it can be used as such. Judging the use of Git and GitHub based on LMS features [20], GitHub supports many

of the important learning features (e.g., assignment delivery and submission), but it also lacks certain features (e.g., grading management tools). In examining the model published by Malikowski *et al.* [25], we come to similar conclusions, where platforms like GitHub have the capability to support all five categories of the model. However, student and course evaluation may require additional work, as well as developing programs to detect and automatically grade student pushes.

Lane [22] discusses two sides of the CMS. From one side, CMS are a **toolbox** for educators, providing a default course structure, and managerial and administrative features. From the other side, integrated commercial CMS are a **trap** that limit faculty creativity, are difficult to customize (sometimes this involves additional cost), and they make it hard to accommodate individual teaching styles. As GitHub is adopted for more and more courses, it remains to be seen if it suffers from similar or different issues to mainstream CMSs.

Going Beyond Traditional LMSs

Beyond the features discussed above, GitHub provides a number of other tools and features that allow educators more novel possibilities.

Version Control

According to the interviewees that teach technical courses (such as computer science), students destined for careers in software and information technology benefit from learning to use version control systems [3]. This reflects the benefits mentioned in previous work on these types of systems [30]. At the same time, version control allows educators to see the final results as well as the processes students used to produce the results [13]. However, this is a double-edged sword as the technical aspects of version control, specifically Git, is one of the main challenges mentioned by interviewees.

Transparency & Awareness

Kreijns *et al.* [19] discuss the shortcomings of contemporary CSCL, particularly in group learning, social construction of knowledge, and the learning process itself. They link these problems to the impeded social interaction of CSCL environments caused by two major pitfalls: taking social interaction in groups for granted, and the lack of attention paid to the social psychological dimension of social interaction outside of the task context. **Transparency of activities** allows GitHub to address this by embedding **social affordances** and by supporting **group awareness**, as proposed by the suggested theoretical framework [18].

However, only a few of our participants mentioned using the **transparency** and **awareness** features to promote collaboration among students. If taken advantage of, GitHub's transparency could be beneficial for collaborating students [7], creating awareness of each others' activities that could, in turn, support collaborator creativity [11]. As such, GitHub is a good example of a transition from a space to a place, as it transitioned from a hosting service that provided space for projects, to a place where work, activities, ideas and discussion can be shared [8].

Student Engagement & Participation

GitHub enables students to contribute to course material or work done by other students with the use of the Pull Request mechanism—a novel way to participate in courses. GitHub provides more in-depth ways to communicate and collaborate and provides teachers and students with a way to gain social group awareness and get information about what group members are doing, who they are communicating with, and how they are contributing [16]. Thus, GitHub enables a participatory culture [17] where students can create, edit, and share in such a way that they feel their contributions matter.

Indeed, one of our interview questions asked whether the use of GitHub encouraged student participation. Impact on student engagement and participation came up several times by the interview participants. However, except for some specific examples, the interviewees didn't have any metrics to measure this. In that sense, our study is limited and leads to future work to investigate if GitHub encourages student participation.

Reusing and Remixing

Some interviewees described the ability to pull from other sources or old material as useful for creating and organizing course material. The main way to reuse materials as seen from our findings was to fork an old instance of a course. However, there were also instances of educators collaborating with each other to create course material, or sharing material with instructors in other universities so that material can be remixed as an instructor sees fit. With current learning management tools, as educators ourselves, we have noted it is extremely tedious to version and share course materials. We end up with each new course as a new entity, losing the history of where we got the materials and who contributed to them. GitHub mitigates this through the use of the *Blame* feature, which visualizes the providence of file changes. In time, we expect to see large networks of educators contributing to the same course, and know who contributed to the material and how.

Implications Beyond GitHub

Our findings go beyond GitHub and apply to related environments, such as BitBucket[27], that provide similar social and collaborative features. In fact, a few of our interviewees also mentioned using BitBucket and described similar experiences using that tool. The lessons learned can be applied to other social hosting tools as well.

Recommendations

A question that an educator might ask at this point is, *"is GitHub ready for 'prime-time,' or should it only be used by early adopters for now?"* We feel that the answer currently lies somewhere in between. From one aspect, GitHub's current platform is considered mature, supports many learning features, provides numerous benefits in the educational context, and is available to anyone wishing to try it out (for free). On the other hand, the use of GitHub involves several challenges (as reported in our Findings, but not repeated in the

[27] `https://bitbucket.org/`

follow-up survey), and currently, is not widely used among educators. As the environment evolves and educators learn and share more about using the platform, we expect that some of these challenges, such as the steep learning curve, may be mitigated.

Our recommendation to educators is to share their experiences and resources with other educators. This shared information should be organized in a shared knowledge base, created by GitHub, and maintained by the community. Furthermore, GitHub (or a third-party developer) can further resolve some of the challenges by tailoring the platform for educational purposes (e.g., providing support for additional file formats). Once these changes are made, we expect to see a rapid adoption of GitHub by educators.

Limitations

From the data collection aspect, we collected data from the educator's point of view only. However, the student's point of view is also crucial and may provide additional support to our findings or reveal new insights. Additionally, the participants of our study were recruited under the condition that they already use GitHub to support learning, where it may imply that their experience had to be (somewhat) successful. This means that our study may have missed the challenges faced by potential participants who failed to use GitHub to support teaching. Additionally, our participants used GitHub at different periods of time, thus their experiences may differ and this may affect our findings.

CONCLUSIONS

In this study, we found that GitHub can be a powerful learning management tool, and while it provides many benefits, there are challenges involved as well. Our findings indicate that various educators use GitHub differently, even in similar environments (e.g., technical background) and with similar requirements (e.g., class size, course type). In some cases, this may happen due to differences in requirements or personal preferences. However, in other cases, there are different uses because GitHub's potential in the context of education is still emerging and many educators aren't aware of the possible uses and advantages it may bring. In this sense, suggested practices should be formed and shared among educators to better utilize GitHub's capabilities as a collaboration platform.

The contribution of this research is fourfold: (1) insights into how GitHub is being used in educational contexts; (2) a list of the benefits that GitHub's content hosting and social features provide in education; (3) a description of the challenges and problems related to using GitHub for education; and, (4) recommendations and suggested practices for how GitHub could be used to support learning. The findings from this research shed light on the use of GitHub as a collaborative social learning platform, and show that tools such as GitHub have the potential to bring more collaboration and transparency to education.

Lastly, this is only the first step in investigating this phenomena. The next steps should be to investigate the student perspective in using GitHub to support learning, and to conduct in-depth case studies of how GitHub plays a role in specific courses.

ACKNOWLEDGMENTS

We thank Cassandra Petrachenko for her valuable comments and editing support. We are extremely grateful to all the interview and survey participants. We would also like to thank GitHub for their willingness to help with this study.

REFERENCES

1. Alhazmi, A., and Rahman, A. Why lms failed to support student learning in higher education institutions. In *E-Learning, E-Management and E-Services (IS3e), 2012 IEEE Symposium on* (2012), 1–5.
2. Begel, A., Bosch, J., and Storey, M.-A. Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder. *Software, IEEE 30*, 1 (2013), 52–66.
3. Britton, J., and Berglund, T. Using version control in the classroom. In *Proceeding of the 44th ACM technical symposium on Computer science education*, ACM (2013), 753–753.
4. Clifton, C., Kaczmarczyk, L. C., and Mrozek, M. Subverting the fundamentals sequence: Using version control to enhance course management. *SIGCSE Bull. 39*, 1 (Mar. 2007), 86–90.
5. Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. Social coding in github: Transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, ACM (New York, NY, USA, 2012), 1277–1286.
6. Dalsgaard, C. Social software: E-learning beyond learning management systems. *European Journal of Open, Distance and E-Learning 2006*, 2 (2006).
7. Dalsgaard, C., and Paulsen, M. F. Transparency in cooperative online education. *The International Review of Research in Open and Distance Learning 10*, 3 (2009).
8. Dourish, P., and Bellotti, V. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, ACM (1992), 107–114.
9. Downes, S. Feature: E-learning 2.0. *Elearn magazine 2005*, 10 (2005), 1.
10. Easterbrook, S., Singer, J., Storey, M.-A., and Damian, D. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer, 2008, 285–311.
11. Farooq, U., Carroll, J. M., and Ganoe, C. H. Supporting creativity with awareness in distributed collaboration. In *Proceedings of the 2007 international ACM conference on Supporting group work*, ACM (2007), 31–40.

12. Gikas, J., and Grant, M. M. Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones & social media. *The Internet and Higher Education 19* (2013), 18–26.
13. Glassy, L. Using version control to observe student software development processes. *Journal of Computing Sciences in Colleges 21*, 3 (2006), 99–106.
14. Griffin, T., and Seals, S. Github in the classroom: Not just for group projects. *J. Comput. Sci. Coll. 28*, 4 (Apr. 2013), 74–74.
15. Grosseck, G., and Holotescu, C. Can we use twitter for educational activities. In *4th international scientific conference, eLearning and software for education, Bucharest, Romania* (2008).
16. Janssen, J., and Bodemer, D. Coordinated computer-supported collaborative learning: Awareness and awareness tools. *Educational Psychologist 48*, 1 (2013), 40–55.
17. Jenkins, H., Clinton, K., Purushotma, R., Robison, A. J., and Weigel, M. Confronting the challenges of participatory culture: Media education for the 21st century, 2006.
18. Kreijns, K., Kirschner, P. A., and Jochems, W. The sociability of computer-supported collaborative learning environments. *Educational Technology & Society 5*, 1 (2002), 8–22.
19. Kreijns, K., Kirschner, P. A., and Vermeulen, M. Social aspects of cscl environments: A research framework. *Educational Psychologist 48*, 4 (2013), 229–242.
20. Kumar, S., Gankotiya, A., and Dutta, K. A comparative study of moodle with other e-learning systems. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 5 (April 2011), 414–418.
21. Lacey, A., and Luff, D. *Qualitative data analysis*. Trent Focus Sheffield, 2001.
22. Lane, L. M. Toolbox or trap? course management systems and pedagogy. *Educause Quarterly 31*, 2 (2008), 4.
23. Liaw, S.-S., Chen, G.-D., and Huang, H.-M. Users attitudes toward web-based collaborative learning systems for knowledge management. *Computers & Education 50*, 3 (2008), 950 – 961.
24. Majumder, A., Datta, S., and Naidu, K. Capacitated team formation problem on social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, ACM (New York, NY, USA, 2012), 1005–1013.
25. Malikowski, S. R., Thompson, M. E., and Theis, J. G. A model for research into course management systems: Bridging technology and learning theory. *Journal of educational computing research 36*, 2 (2007), 149–173.
26. McLure Wasko, M., and Faraj, S. it is what one does: Why people participate and help others in electronic communities of practice. *The Journal of Strategic Information Systems 9*, 2 (2000), 155–173.
27. Nickerson, R. S., and Zodhiates, P. P. *Technology in education: Looking toward 2020*. Routledge, 2013.
28. Pappano, L. The year of the mooc. *The New York Times 2*, 12 (2012), 2012.
29. Perez De Rosso, S., and Jackson, D. What's wrong with git?: a conceptual design analysis. In *Proceedings of the 2013 ACM international symposium on New ideas, new paradigms, and reflections on programming & software*, ACM (2013), 37–52.
30. Reid, K. L., and Wilson, G. V. Learning by doing: Introducing version control as a way to manage student assignments. *SIGCSE Bull. 37*, 1 (Feb. 2005), 272–276.
31. Reinhardt, W., Wheeler, S., and Ebner, M. All i need to know about twitter in education i learned in kindergarten. In *Key Competencies in the Knowledge Society*. Springer, 2010, 322–332.
32. Seaman, C. B. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on 25*, 4 (1999), 557–572.
33. Storey, M.-A., Singer, L., Cleary, B., Figueira Filho, F., and Zagalsky, A. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, ACM (2014), 100–116.
34. Sultan, N. Cloud computing for education: A new dawn? *International Journal of Information Management 30*, 2 (2010), 109–116.
35. Thompson, C. How khan academy is changing the rules of education. *Wired Magazine 126* (2011).
36. Tsay, J. T., Dabbish, L., and Herbsleb, J. Social media and success in open source projects. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, CSCW '12, ACM (New York, NY, USA, 2012), 223–226.
37. Wakeford, N., and Cohen, K. Fieldnotes in public: using blogs for research. *The Sage handbook of online research methods* (2008), 307–326.
38. Wu, Y., Kropczynski, J., Shih, P. C., and Carroll, J. M. Exploring the ecosystem of software developers on github and other platforms. In *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW Companion '14, ACM (New York, NY, USA, 2014), 265–268.