



# LUND UNIVERSITY

## Enhancing Student Engagement Using GitHub as an Educational Tool

Gunnarsson, Sara; Larsson, Peter; Månsson, Sara; Mårtensson, Erik; Sönnerup, Jonathan

2017

*Document Version:*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Gunnarsson, S., Larsson, P., Månsson, S., Mårtensson, E., & Sönnerup, J. (2017). *Enhancing Student Engagement Using GitHub as an Educational Tool*. Genombrottet, Lunds tekniska högskola.  
<http://journals.lub.lu.se/index.php/KG/issue/archive>

*Total number of authors:*

5

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00



**LUND**  
UNIVERSITY

# Enhancing Student Engagement Using GitHub as an Educational Tool

Sara Gunnarsson, Peter Larsson, Sara Månsson,  
Erik Mårtensson and Jonathan Sönerup\*



2017-06-20

---

\*The list of authors is in alphabetical order. The picture is of the professor version of the Octocat, the GitHub logo, which was retrieved on 2017-06-16, from [https://octodex.github.com/Professortocat\\_v2](https://octodex.github.com/Professortocat_v2).

### **Abstract**

Student engagement is an important factor when it comes to the learning process. Students who do not feel engaged in their studies are more likely to fail their exams and might even drop out of their education. Different methods have been tried in order to increase student engagement. One of them is the contributing student pedagogy, where students contribute to the learning of others and also value the contributions of others. To facilitate this pedagogy different learning management systems can be used. One of these tools is GitHub, which is a social and collaborative environment, traditionally used for software development. This study further investigates the possibilities of using GitHub in education in terms of a literature and case study. Results show that using GitHub in education can lead to increased student engagement, if used in a proper manner. In conclusion, GitHub is a prominent tool with lots of potential to be used in and for educational purposes.

**Keywords:** Student engagement, Contributing student pedagogy, Learning management systems, Educational tools, GitHub

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Student Engagement</b>	<b>1</b>
2.1	Student Motivation . . . . .	2
2.2	Authenticity in Learning . . . . .	2
2.3	Feedback . . . . .	3
2.4	Deep Approach to Learning . . . . .	3
<b>3</b>	<b>Contributing Student Pedagogy</b>	<b>3</b>
3.1	Technology Supporting the Contributing Student Pedagogy . . . . .	4
<b>4</b>	<b>Learning Management Systems</b>	<b>5</b>
<b>5</b>	<b>Git and GitHub</b>	<b>6</b>
5.1	GitHub for Education . . . . .	7
<b>6</b>	<b>Studies on GitHub Usage in Education</b>	<b>7</b>
<b>7</b>	<b>Case Study</b>	<b>8</b>
<b>8</b>	<b>Discussion</b>	<b>9</b>
<b>9</b>	<b>Conclusion</b>	<b>10</b>
	<b>References</b>	<b>11</b>
<b>A</b>	<b>Interview with Björn Regnell</b>	<b>13</b>

## 1 Introduction

A question which always is very relevant to discuss concerning education is how to increase the student engagement. There are several factors related to this topic, for example the students' motivation, the authenticity in learning, feedback during the course, and these factors are all related to the students' deep learning process. One pedagogy which has been shown to increase student engagement and stimulate the students' deep learning process is the contributing student pedagogy, which focuses on the students' contributions to the learning of other students and also that they value the contributions of others [10].

In the era of digitalization the society is changing and more tools and platforms are becoming available to a larger population. This change includes the learning environment and it opens up for new possibilities for people to share knowledge and to collaborate with each other using publicly available platforms.

One of these tools, GitHub, was originally developed as a social and collaborative tool for software development. It is a platform which encourages people using it, to share their work and also to give feedback and help to improve the work of others. Even though the original purpose of GitHub was for software development, lately, another use of GitHub has emerged, namely using GitHub in education as a learning management system (LMS). The social and collaborative environment can be exploited in many ways in a course.

After this introduction, the paper starts with a literature study in Section 2, which first elaborates on the concept of student engagement and the specific aspects of it, which are focused on in this report. Then in Section 3, the literature study focuses on the contributing student pedagogy, followed by an introduction to LMS in Section 4, and then Git and GitHub in Section 5. The literature study ends in Section 6, which investigates the situations and contexts where GitHub has been used as a tool in education until now. The second part of the report, found in Section 7, is a case study, in order to relate theory to an example, of a course at Lund University, which was developed using GitHub. Finally, Section 8 discusses the results found in the literature and case studies and Section 9 concludes the paper.

## 2 Student Engagement

Student engagement has been defined as “*the amount of physical and psychological energy that a student devotes to the academic experience*” [2]. This amount can be affected by a number of different factors, which may be derived from different aspects of a student's life. There are many social factors, which have been proved to have an impact on a student's engagement, e.g., whether the student lives in a student dormitory or not, or if the student takes part in campus based social activities in his/her spare time [2]. These are factors which the educational institution may or can not affect, but there are a lot of academic factors which the institution have a large influence on. Among other things, it has been proved that important factors are to have enthusiastic and engaged teachers [6], and effective educational practices [15]. The latter has been defined by Kuh [14], who introduced five benchmarks: level of academic challenge, active and collaborative learning, student interactions with faculty members, enriching educational experiences, and supportive campus environment. These benchmarks require that the educators put a lot of effort into creating

a good teaching climate, which motivates the students to invest both time and energy into their studies - otherwise they will feel less engaged and might even drop out of school before finishing their studies [2]. There are several ways of creating student engagement, that can be used to create a genuine educational interest among the students. This report will focus on student motivation, deep approach to learning, feedback between teachers and students, and authenticity in learning.

## 2.1 Student Motivation

Throughout their education, students will experience periods with different levels of motivation. The challenge of the teacher is to create motivation amongst the students at the beginning of their education or at the beginning of a new course, in order to create an interest and a curiosity about a new discipline. In order to do so, it is important that the students experience that what they do is valuable, and that they have a chance of succeeding when engaging the task, or course, at hand [3]. If the students do not experience this, they will most likely not feel motivated.

Motivation can be divided into two different types of motivation: *extrinsic* and *intrinsic*. Extrinsic motivation is affected by factors outside of the individual, such as getting higher grades in order to get a better job in the future, or making their parents happy. Intrinsic motivation comes from within the individual, and can be described as each individual's curiosity and interest to learn something new that they will benefit from in the future [6]. Most teachers are interested in inducing intrinsic motivation among their students, since this generally creates a larger extent of engagement and makes the students more apt to persist with their education [18]. The teacher will have to identify different techniques to reach the intrinsic motivation, which poses a challenge with each new student group since each group is different and contains students with different levels of interest in a subject. In order to handle this problem, it is important that the teacher is interested in understanding, and help solving, the challenges that each individual student is facing when starting a new course. Each student will have a different set of knowledge and each student will have different preconceptions about the course as well. The problem that the teacher will have to solve is to find ways to help their students create new knowledge based on what they already know, instead of trying to push new information onto the students [6]. If the teacher is able to help the students create knowledge themselves by generating curiosity and interest, the students will be able to create individual, comprehensive pictures of the subject which originates from themselves [6]. This will create a greater understanding of the relevance of the new knowledge, which will also trigger their intrinsic motivation to learn more about the new subject [6].

## 2.2 Authenticity in Learning

In order to further create an interest and close relation between the students and the subject, it is important that the teacher acquires a certain level of authenticity in the education. Authenticity in learning can occur in different forms: a teacher can show authenticity by being genuine and self aware, a teacher can create important connections between students and teachers by being authentic, and an authentic teacher can also create interest amongst the students and make them enthusiastic about the subject [13]. The material of a course can also be authentic, which means that the material is written for the students by their peers. An example of authentic material is texts written in English by native English speakers for native English speakers [12]. This example

could be extended to include course material for a course written by students who have already taking the course. By including the older students, it is possible to show the new students what they are expected to know after the course and the information will be communicated to them in a way that, probably, is more understandable since it is written by older students who have recently taken the same course. This will bring the subject and the course material closer to the students and will create a more relatable teaching environment.

## 2.3 Feedback

In order to make sure that the students are grasping the contents of a course the teacher has to be perceptive to the students' thoughts and ideas. It is also important that the teacher evaluates what and how well the students have learned in order to figure out what is and is not working in the classroom [9]. The purpose of an evaluation could also be to investigate what the students thought about the course when it was completed, but the most important evaluation with regards to student engagement occurs during the course. If the teacher continuously evaluates the students' knowledge and interest in the subject, it is possible to give the students feedback on their work to a much larger extent than if the teacher only performs an evaluation after the course is finished. The most important aspect of the feedback is that it is constructive and developing, and does not only discuss the negative aspects of the students' work [6]. Students who receive continuous feedback about their work are more likely to feel motivated and develop throughout their education [6].

## 2.4 Deep Approach to Learning

If the students feel engaged in a course, it is more likely that they will use deep learning strategies in order to obtain knowledge. Students that use deep learning strategies are trying to understand the purpose and the main message of the course content, instead of only trying to memorize the most important details [6]. The students will also try to connect their new knowledge to what they already know and try to create a unity of their aggregated knowledge [6]. In order to help the students reach deep learning, it is important that the learning activities are focused on teaching for understanding and more importantly personal understanding for each student [16]. This requires that the teacher shows his/her own engagement in the course and subject and that the teacher can relate to the students' interests in, and prior knowledge of, the subject [6]. There are several different ways of doing this, but feedback and continuous assessment is an important aspect of it. It is also important to include different learning activities which may appeal differently to different persons. Studies have shown that learning activities which promotes self and peer assessment such as group projects promotes a deep approach to learning [16], that is, learning activities where the students have to produce something on their own instead of reproducing something [1].

## 3 Contributing Student Pedagogy

The concept of when students are encouraged to contribute to the learning of others and to value the contribution of others is called contributing student pedagogy (CSP) and it is based on social

constructivism and community-based learning. It is defined by Hamer as “*a pedagogy that encourages students to contribute to the learning of others and to value the contributions of others*” [10]. The paper identifies some characteristics of CSP, which are that the students’ roles are shifting from passive to active, there is a focus on contributing to the course content, contributions are assessed in terms of quality, learning communities are developed and student contributions are facilitated by technology in some way.

In relation to student engagement, by shifting the roles of the students and the teacher, the teaching situation becomes more like the reality when working at a company or with research, i.e, the situation becomes more authentic. Further on, to be able to contribute to the course content, a deep approach to learning the material is required. Also, being involved in this kind of learning community, where students receive feedback on what they do and they feel like they are a part of something, can increase their motivation. Although, some students could feel uncomfortable in the situation, which then could lead to decreased motivation.

As a real example, in [7], the authors used the principle of CSP when redesigning the curriculum for their foundational programming courses and other key courses. They found that using the approach of CSP was beneficial in the sense that it increased engagement and participation. It was also beneficial for the development of critical analysis, collaboration and problem solving skills.

Students could also be included in the pedagogical planning and Bovill et al. [4] have identified three ways in which this can be done; the first one is in the teaching approaches, the second one is in the course design and the last one is about the curriculum. Benefits with this are that students and academic staff gain a deeper understanding of learning, they experience enhanced engagement, motivation and enthusiasm and also that students and academic staff relate differently. Challenges with including the students in the pedagogical planning is that the teacher gives up some parts of controlling the pedagogical planning, the time-investment needs to be reconsidered, meeting the professional requirements and gaining access to the students.

### 3.1 Technology Supporting the Contributing Student Pedagogy

Technology which can facilitate contributing student pedagogies should facilitate collaboration and communication, develop necessary attitudes and skills, support the authoring of learning resources and collect data on student activity and performance [10]. A tool supporting contributing student pedagogy activities could be of different nature and some uses, as according to [11], are peer review, dialogue and discussion, annotation, content construction, solution sharing, activity creation and making links. Peer review is simply that students look at other students’ work and make comments. In dialogue and discussions, the contributions come in the conversations between the students. With annotation, it means that students comment on material that already exists, and is not created by students, and thereby the students share their comments with others. Both content construction, solution sharing and activity creation means that the students are creating something, which they share with other students so that they also can learn from it. The concept of making links means that students are linking known concepts together or that they find other sources which somehow is related to the course and share this with the other students, for example at a wiki.

Elaborating on how technology can be used to support a pedagogy, which is contribution-oriented, Collis et al. [5] describes how different components of such a system can be utilized. About the



general course organization, a contribution-oriented approach could be to let the students add links to material related to the course content. The part of the system which supports lectures could also be used so that the students after the lecture can discuss and reflect about the material, also allowing them to add contributions to the material. A component for self-study and exercises in the system could allow students to use other students' submissions, allowing them to learn from these, and then in a structured way give each other feedback and answer each others questions. Regarding multi-session projects or activities, a contribution-orientation could mean that the students have a shared workspace in combination with other communication and reporting tools and in this way they can collaborate on complex problems. In this process the students should also be guided to give each other feedback as the project proceeds by structured communication and by having their partial products accessible for the other students. The assessment aspect of the course could integrate new forms of assessment, such that the students can maintain their own portfolio in the system. This could also include peer-assessment activities. For the general communication in the system a Web-board could be used, where students are responsible for moderating the discussions. Experts from outside of the course could also be included and respond to questions.

## 4 Learning Management Systems

Learning management systems (LMS) are software applications used in educational courses for administration, tracking and reporting. It lets teachers create and distribute course material such as quizzes, hand-ins, projects and much more, usually with a user-friendly interface. Common LMS tools include Moodle<sup>1</sup>, Canvas<sup>2</sup> and Desire2Learn<sup>3</sup>.

In [17], Malikowski et al. developed a model in which the different uses of LMS are divided into five categories, these are:

**Transmitting course content** The main usage for LMS is the dissemination of course content, such as lecture notes, assignments and grades. The research shows that the most frequent case is to make announcements for upcoming deadlines or exams.

**Creating class discussions** There are two ways of communication through an LMS; synchronously and asynchronously. Asynchronous communication is comparable to e-mail, where students usually compose a more thoughtful note or question. The synchronous communication is in real-time, like a messaging system, where the students can interact in a more social way. The research shows that the dominating form of communication in an LMS is asynchronous.

**Evaluating students** The most prominent feature of LMS is the ability to create tests, usually in the form of quizzes. This is a good complement to other types of examination and aids different learning styles.

**Evaluating courses and instructors** An LMS can be used to evaluate the course itself along with the instructors, in the form of quizzes or surveys. There is not much data on the subject to draw any conclusions regarding its usefulness.

---

<sup>1</sup><https://moodle.org/>

<sup>2</sup><https://www.canvaslms.com/>

<sup>3</sup><https://www.d2l.com/>

**Creating computer-based instruction** A computer-based instruction (CBI) is a learning tool in which students interact with a computer. Some examples include; quizzes, where the questions are drawn randomly and gives some feedback; the use of multimedia, such as video, text and audio; interactive material, such as graphs and images.

## 5 Git and GitHub

Git is a Version Control System (VCS) and is used for coordinating work between several parties. The main application is for collaboration when developing software code, but it may be used for any type of files. A Git project is known as a *repository*, or simply *repo*. Git is decentralized, i.e. there is no need for a single host for storage. This allows anyone to have their files version controlled. When collaborating, all collaborators work against a single server hosting the files. This allows the developers to stay synchronized and up-to-date. The command `git clone` will download a copy of the repo from the server to the user's computer, allowing a new developer to make changes and contribute. When a developer has added, changed, or removed material, they simply type `git commit` to add the changes, also known as *commits*. Using the command `git pull` will retrieve the latest updates from the server.

GitHub<sup>4</sup> is an online hosting service for Git repositories. It is free of charge and it is one of the largest hosting services available. GitHub lets you create an account and stores all of your repositories, which you can manage through a user-friendly web interface. GitHub also extends Git and/or makes it easier in several ways:

**Fork** GitHub lets you copy (fork) a repository you do not own to your profile. This allows you to develop your own version of the project. This is common in software and a good example is the different Linux versions that exist, e.g., Ubuntu and Linux Mint are just forks of Debian. They share the same code base but then takes different directions (as a fork).

**Pull request** If you have forked a repository and added some nice features that you would like the original author to include, you make a pull request. This way, the original author will receive a message saying that person X has made changes and wants you to include the changes in the project. If the author accepts the changes, everyone using the project will now have access to the new features.

**Issue** An issue is basically a request to fix a bug or to add a new feature in the code. You can assign an issue to a specific developer or letting all developers be aware of it. When a developer has taken care of the issue, he or she will mark it as fixed and the creator of the issue will be notified.

**Wiki** For every repository there is an optional wiki. In the wiki, the developers can explain how to use the code by providing some tutorials or any other relevant information.

**Visual feedback** GitHub provides the users with graphs of all commits made and by which contributor. This allows anyone to follow the project during its development.

---

<sup>4</sup><https://www.github.com>

## 5.1 GitHub for Education

As stated previously, Git and GitHub is not only used for programming code, but may be used for any type of files, such as course material and code examples, relevant links on the wiki etc<sup>5</sup>. GitHub is so versatile that it can be used in a lot of unexpected ways<sup>6</sup>.

Evaluating GitHub as an LMS, it can be found that GitHub fulfills the requirements of several of the categories in the model developed in [17]. Teachers can easily put lectures notes, assignments and lab instructions on GitHub and thereby transmit course content. Asynchronous class discussions can be made through issues. GitHub has also made it easy to evaluate students, in the sense that the teacher can follow and assess their qualitative and quantitative contributions. To some degree you can also evaluate courses and instructions by making pull requests and creating issues. GitHub does not currently support the creating of computer-based instructions.

## 6 Studies on GitHub Usage in Education

There has been scientific studies made on the usage of GitHub in education. In [8] Feliciano et al., they did a case study on GitHub as an educational tool, by looking at two university courses, from the perspective of the student. One course was in Distributed Systems with both graduate and undergraduate students and one was a Software Engineering course for senior undergraduate students. GitHub was used for a few different things.

First of all, course material was disseminated through a repository, including lab assignments, links for readings and homework. It was possible for students to suggest changes through issues and pull requests, but this possibility was not emphasized during the courses. Secondly, GitHub was used to host lab content and lab related discussions. Finally, GitHub was used for the students to publicly host their individual and group project work. Although this was not mandatory, most students posted their content on GitHub, which made it possible for other students to give feedback on it. The study consisted of voluntary interviews with course participants during the courses and a survey afterwards, that tried to validate the findings from the interviews.

Some benefits of using GitHub, from a student engagement perspective, were found. By using GitHub the students learned an industry-relevant tool, which made the course more authentic. GitHub simplified cross-team collaboration and contributions. Some labs required that students looked at the code from other groups, but many students often looked at the codes of others more than that. In some cases students utilized the codes of other groups and thereby found and fixed bugs. Finally, GitHub made it possible for students to make contributions to the course material through issues and pull requests. It should be noted that while this option was not used that much, it would probably have been used much more if it was emphasized more and if the suggestions for changes were treated faster.

In [19], Zagalsky et al. did a three part study, from the lecturer's perspective, on GitHub used as an educational tool. First they started with an exploratory phase, where they looked for different uses of GitHub by educators, which helped them in creating useful research questions. Then lecturers

---

<sup>5</sup><https://classroom.github.com/>

<sup>6</sup><http://readwrite.com/2013/11/08/seven-ways-to-use-github-that-arent-coding>

from 15 different Computer Science, Sciences and Humanities courses were interviewed. Finally, the researchers made a validation survey.

The educators used GitHub as a submission platform and to host course content. Also, in this study some benefits from a student engagement perspective were found. Just as in [8], it was noted that the tool encourages student participation by making it easy for students to suggest changes via issues and pull requests. The fact that the suggestions stay recorded forever further rewards student participation. Furthermore, participation was encouraged in an indirect way, for example by teachers using students logs as material for discussion.

The fact that who made each of the steps of the progress on a group project on GitHub was visible to everyone, including the teacher, incentivised actually working hard on the project and penalized free-riders. GitHub made it easy to share material both between students and between teacher and students. Having a public repository even made it possible for people from outside the course to contribute to the material. Also in this study it was pointed out that students became motivated by the fact that GitHub is relevant to know in industry and having their material available on GitHub makes it easy for students to share it with a potential future employer.

## 7 Case Study

In order to create a closer connection between theory and practice, this section includes a case study of a course where the teacher used GitHub in order to develop new course material. The study consists of an interview with the lecturer, Björn Regnell, which can be found in Appendix A. This section describes the relation to the theory presented in the previous sections.

The course which has been examined is an introductory programming course, which is mandatory for first-year computer science engineering students, and eligible for students at the environmental engineering program at the Faculty of Engineering (LTH), at Lund University. The course took place during the fall of 2016 and the number of students registered on the course at the course start was 125. The course responsible was Björn Regnell, professor in Software Engineering at the Department of Computer Science at LTH. Prior to the course start, the course was redesigned in order to make the course more evenly challenging for the entire student group. The old course content led to that some students with a lot of prior programming knowledge were understimulated during the course, while some students that were not accustomed to programming before starting, experienced that the course was very difficult to follow and pass. The redesign of the course meant that they switched main programming language from Java to Scala, and that entirely new course material was developed before the course start. It also meant a change in the way the students collaborated throughout the course, where student groups consisting of students with different amounts of programming experiences were created. The purpose of these groups was to give all students a challenge during the course. The more experienced students were supposed to help and explain things to the more inexperienced students during study sessions. This promotes deep learning and student engagement, since the students have to help each other by giving feedback on the work created in the course as well as working together in some aspects to reach the best solution to the learning problems of the course.

The new course material was developed together with older students, who had taken the course previous years in its old form. The material was developed using GitHub, where the contributors

could use all of the common GitHub functions in order to develop the material. Each one of the older students was given a part in the project, either as a leader or as a developer. They could then declare an interest in taking part in different tasks of the material development, after which Regnell decided which students who were supposed to work with which task. The contributing students were given a salary based on which task they contributed to and when the tasks were finished. Regnell experienced that the contributing students really enjoyed working with the course material and that they learned a lot throughout the project, and many of the contributing students also decided to take part in the course during the semester, as lab and exercise instructors. Regnell also experienced that:

*“It became sort of a partner ownership among the computer science students at LTH for this introductory course, which I find very delightful. And this feeling has affected the students at the course”.*

Regnell’s experience is an example of when material created for students by their peers creates a larger authenticity in the material. The course content becomes easier to comprehend since the students of the course have a relationship with the material via the older students. This also creates a closer connection between the students and the subject, implying that the deep learning of the students is easier to obtain.

The largest challenge Regnell faced during the development of the course material was to get material of good enough quality. In order to do so, he had to go through each individual pull request and make sure that the new content was relevant. It was also important that the course material helps reaching the learning objectives of the course, in order to create a cohesive image of the course objectives, learning activities and course examination.

Even though the course material was finished before the start of the course, it was available on GitHub during the course. This meant that the students at the course were able to contribute to the material in the same way as the older students. Regnell’s experience was that only the more advanced students, who had already used GitHub before starting the course, were doing this. This meant that the student engagement of contributing to the course material during the course was not very large, but Regnell did not expect it to be either. He stated during the interview, that the objectives of using GitHub during the course was not to use it as a software development tool, but instead to provide the more advanced students with the possibility to contribute to the course. For these students, this could make them feel more motivated throughout the course since they had more challenges available, which may maintain their interest in the course to a larger extent.

## 8 Discussion

Throughout this study GitHub has shown to fulfill a lot of the requirements of an LMS, such as transmitting course content, enabling asynchronous communication, making it possible to evaluate students in terms of contributions and enabling immediate feedback on the course content.

GitHub can also be used as a technology to support a CSP in many different ways. It can for example facilitate collaboration and communication both in group projects and individual projects, as well as about the course content. This leads to increased feedback opportunities and that a

learning community develops, which can increase both the motivation among the students, as they see their role in a larger context, and the deep learning process, as they learn from each other.

There are many ways to use GitHub within education. Depending on how GitHub is used, it can lead to different pedagogical benefits. One application, as seen in the case study, is to use GitHub for course development by older students. For the older students developing the course, this could lead to a feeling of authentic learning as they are working on a platform which is used in industry. Being able to contribute to the course content also requires a deep approach to learning among the students. For the students taking the new course, the course material is experienced as more authentic since it is developed by older students and hopefully it will lead to increased deep learning as well. This way of using GitHub when developing a course also allows for immediate feedback among the developers. One difficulty with letting older students contribute to a course is guaranteeing a certain level of quality of the material.

Another use for GitHub in education is to allow students to contribute to the course material in general during a course. This is in line with the CSP and can be used to encourage different learning styles which can lead to increased student engagement. Since all students learn in different ways, this way of using GitHub could lead to deep learning among a bigger group of students since several learning styles are supported. In relation to this, the intrinsic motivation could increase as students are encouraged to learn in their own way. Challenges with letting students contribute to the course material are the quality of the material as well as the fact that the contributions have to be moderated in some way and with a lot of contributions, this could be time consuming.

GitHub could also be used to store individual and group projects, allowing for immediate feedback and related discussions. This could help improving the student engagement since students are more likely to feel motivated and develop throughout their education when receiving this kind of feedback. Depending on the task, the work can be chosen to be either public or private. Positive aspects of having it public are that future employers can see the work and this might encourage the students to do their best. A possible drawback is that the students might not feel comfortable with having it public and therefore their motivation could decrease [8].

In general, challenges using GitHub in education could be that not all students are familiar with the tool and therefore it takes focus from the actual subject. Also, as previously mentioned, integrity could be an issue if it is public.

## 9 Conclusion

When using GitHub in education, one has to think about the purpose, what the goal is and then how the features in GitHub can be used to reach this goal. GitHub can be used in many different ways, but it might not be applicable in all types of courses because a certain amount of knowledge about Git is required to be able to use the features. However, studies have shown that using GitHub can increase student engagement if used in a proper way.

Using GitHub in education is still quite new, but it has a lot of potential for future applications in education. Also, GitHub is currently working on how they could further support educational purposes. The many benefits of using GitHub, such as facilitating collaboration and feedback, makes it a prominent tool to incorporate in courses.

## References

- [1] Maria Andersson, Jenny Loberg, and Jenny Yngvesson. Autentiska examinationsformer och studenternas upplevelse av stress och inläring vid olika examinationsformer. Internal document, Diarienummer Sveriges lantbruksuniversitet ua 30-3899/09.
- [2] Alexander W. Astin. Student Involvement: A Development Theory for Higher Education. *Journal of College Student Development*, 40(5):518–529, 1999.
- [3] John Biggs and Catherine Tang. *Teaching For Quality Learning At University*. SRHE and Open University Press Imprint. McGraw-Hill Education, 4th edition, 2011.
- [4] Catherine Bovill, Alison Cook-Sather, and Peter Felten. Students as Co-Creators of Teaching Approaches, Course Design, and Curricula: Implications for Academic Developers. *International Journal for Academic Development*, 16(2):133–145, 2011.
- [5] Betty Collis and Jef Moonen. *The Contributing Student: Learners as Co-Developers of Learning Resources for Reuse in Web Environments*, pages 49–67. Springer Netherlands, Dordrecht, 2006.
- [6] Maja Elmgren and Ann-Sofie Henriksson. *Universitetspedagogik*. Studentlitteratur AB, Lund, 3rd edition, 2016.
- [7] Katrina Falkner and Nickolas J.G. Falkner. Supporting and structuring contributing student pedagogy in Computer Science Curricula. *Computer Science Education*, 22(4):413–443, 2012.
- [8] Joseph Feliciano, Margaret-Anne Storey, and Alexey Zagalsky. Student Experiences Using Github in Software Engineering Courses: A Case Study. In *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, pages 422–431, New York, NY, USA, 2016. ACM.
- [9] Center for Teaching and Learning. Do You Know Where Your Students Are? Classroom Assessment and Student Learning. *Speaking of Teaching - Stanford University Newsletter on Teaching*, 4(2), 1993.
- [10] John Hamer, Quintin Cutts, Jana Jackova, Andrew Luxton-Reilly, Robert McCartney, Helen Purchase, Charles Riedesel, Mara Saeli, Kate Sanders, and Judithe Sheard. Contributing Student Pedagogy. *SIGCSE Bull.*, 40(4):194–212, November 2008.
- [11] John Hamer, Helen C. Purchase, Andrew Luxton-Reilly, and Judithe Sheard. Tools for Contributing Student Learning. In *Proceedings of the 2010 ITiCSE Working Group Reports*, ITiCSE-WGR '10, pages 1–14, New York, NY, USA, 2010. ACM.
- [12] David Heitler. *Teaching with Authentic Materials*. Pearson Education, 2005.
- [13] Carolin Kreber, Monika Klampfleitner, Velda McCune, Sian Bayne, and Miesbeth Knottenbelt. What Do You Mean By "Authentic"? A Comparative Review of the Literature on Conceptions of Authenticity in Teaching. *Adult Education Quarterly*, 58(1):22–43, 2007.
- [14] George D. Kuh. What We're Learning About Student Engagement from NSSE: Benchmarks for Effective Educational Practices. *Change: The Magazine of Higher Learning*, 35(2):24–32, 2003.

- [15] George D. Kuh, Ty M. Cruce, Rick Shoup, Jillian Kinzie, and Robert M-Gonyea. Unmasking the Effects of Student Engagement on First-Year College Grades and Persistence. *The Journal of Higher Education*, 79(5):540–563, 2008.
- [16] Raymond Lynch, Patricia Mannix McNamara, and Niall Seery. Promoting deep learning in a teacher education programme through self- and peer-assessment and feedback. *European Journal of Teacher Education*, 35(2):179–197, 2012.
- [17] Steven R. Malikowski, Merton E. Thompson, and John G. Theis. A Model for Research into Course Management Systems: Bridging Technology and Learning Theory. *Educational Computing Research*, 36(2):149–172, 2007.
- [18] Richard M. Ryan and Edward L. Deci. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1):54 – 67, 2000.
- [19] Alexey Zagalsky, Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, and Weiliang Wang. The Emergence of Github As a Collaborative Platform for Education. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW '15*, pages 1906–1917, New York, NY, USA, 2015. ACM.



## A Interview with Björn Regnell

This is a summary of the interview held by the project group with Björn Regnell, regarding his use of GitHub for developing course material. First of all, Björn got the question about how he uses GitHub in his course. He explained that the course needed a big reformation including exercises, lectures and also a new programming language. The reason for the reformation was that the department wanted to address the big difference in pre-knowledge among computer science students and saw pedagogical advantages with changing programming language. Also, the former students, taking the course, had been giving the feedback that they wanted the course to be more modern and challenging. The plan for the reformation started already when the last course was given in its original form and with a format of an open development program. The main reason why GitHub was chosen instead of GitLab, Bitbucket and other platforms was that it is the largest and most visible around the world and Björn pointed out that being a part of this development project could be a valuable experience for the student to put in their future CV. GitHub has emerged as a platform “to show what I am capable to produce” among software engineers and developers. Another reason that also is important for the students taking the course is that they can recognize the older students being a part of the course development - *“It became sort of a partner ownership among the computer science students at LTH for this introductory course, which I find very delightful. And this feeling has affected the students at the course”*.

The project’s budget included, in addition to senior staff, also 70 000 SEK for employing students as assistants in course development. So Björn invited former students six months before the deadline when the new course was held in its new form. The complete project included a new GitHub page ([github.com/lunduniversity/introprog](https://github.com/lunduniversity/introprog)) and the purpose for the former students was to develop labs, answers to the exercises and also test the material provided by the development group. In total 20-30 students were involved in the project, divided into both leaders and developers. The leaders were both development leaders and test leaders, managing developers and testers. So in total four different roles were included in the project. The project plan was divided into different tasks, where one example was developing a lab. When a task was completed a fixed amount was given to the different contributors for that particular task. So the complete budget was divided into all sub-tasks with different fixed rewards and the contributors assigned themselves into the sub-tasks they wanted to be involved in, and depending on their experience it was decided whether they got the position or not. The total individual salary summarized over the personal engagement in completing the tasks ranged from 540 SEK up to 6750 SEK.

Evaluating the project, many of the contributors said that it was *“great fun and I’ve learned a lot”*. This engagement also continued during the course, where many contributors worked as teaching assistants. According to Björn, this project was one of the largest and the most difficult projects that he has been managing during his career as a teacher at the university.

The next part of this interview was about how the students, taking the course in its new form, were using GitHub and how it was introduced in the course. Björn has pointed out that the intention is not to use GitHub as an education tool in that sense, but giving the students that already knew GitHub the opportunity to use it to find the course material, exercises and lab tasks. He introduced that GitHub is available for the course in the introduction week so the students with experience could start to investigate the features on their own. Also, during one of the first lectures, a short introduction to how GitHub works with forks, how to clone a version and make a pull request on

the suggested change or corrected flaws, was made. There is also a short written section explaining how it works. During the exercises the teacher assistants got some questions regarding GitHub from the interested students.

On the question if there are any pros and cons with using GitHub for this application, Björn tried to keep it on a level such that all pull requests could be taken care of and answered. He thinks that he succeeded with that and sees no flaws with this. He pointed out the difficulty of getting enough quality course material during the project and that this is true in general for developing new course content. It needs to be complete so that the teaching aspects of it fulfill all the teaching goals, and putting a bit of that burden on students that do not fully have the aspects of an experienced teacher in the subject is difficult. So Björn said that if he needed to point out one thing that he wanted, it was another senior that could help him going through the pull requests and review the content towards the goals. Using GitHub in the course with students that use the material and sometimes send in pull requests with corrections of spelling and answers, but also tips of useful links, has only been a positive feature.

In the last part of the interview Björn got questions about if he got any feedback from the students regarding using GitHub during the course. Björn answered that he did not specifically ask in the course evaluation form questions about using GitHub as a tool for education, but the general feedback was that the course in its new form was a success. Björn also pointed out that one thing he really appreciated was that most of the students, regardless of their knowledge before they started the course, felt that this new course was challenging. One of the main reason for this success was the collaboration teams with students selected based on their previous programming knowledge. So the good students worked together with weaker students to share knowledge within the group itself. Björn got the question if he thinks GitHub can be a tool to use in the teams. His answer was that it depends, but it is less likely in this course. If so, the majority of the students need to know GitHub beforehand, so that the focus can be on the course itself, rather than use the time in the beginning to learn GitHub, and he specifically pointed out that *“It is very important to develop basic knowledge in programming”* and *“If I use GitHub in my courses, I want them to learn GitHub, but I have chosen to focus on other learning goals”*. Björn did mention though, that he might start to give lectures for those who wants to learn how to use GitHub and he also said that he will introduce a new type of programming lab in the next round of the course that is completely different from what he has used before. This lab does not have one solution, which is the normal with today’s laboratory practicals, and the purpose is to practice different concepts. Since the purpose is not to develop a specific solution, but to describe how you have practiced the different concepts, GitHub or the more local GitLab can be a good tool to use.