Using GitHub to improve the provision of feedback in programming classes

Purpose

This learning technology innovation introduced GitHub into the CodeLab I (CL1) and CodeLab II (CL2) modules of Creative Computing. The core purpose was to improve the provision of feedback and monitoring of student progress. The rationale for this is discussed next.

Rationale

The teaching and learning of programming is widely regarded as being difficult (Robins, Rountree & Rountree, 2003). Novice programmers struggle with the abstract nature of concepts and find it difficult to understand how to combine concepts together in new contexts (Lahtinen, Ala-Mutka and Jarvinen, 2005). This may be due to the way programming course are taught, with new concepts introduced each week. However, a poor understanding of previously taught materials can lead to weaknesses constructing knowledge of new concepts (Saeli et al., 2011).

Feedback is one of the most important methods of correcting poor understanding and aiding student learning. Without feedback students are unable to rank their development and may continue to face difficulties (Jonsson, 2013). Research has shown that for feedback to be effective feedback should be is timely, task specific, corrective and seen as an open dialogue (Wolsey, 2008; Jonsson 2013; Nicol and MacFarlane-Dick, 2006). GitHub was chosen as it attends to each of these factors and thus with its use the aims were to enhance the tracking of progress and provision of feedback that extends in class support.

Implementation

GitHub was implemented by providing students with a module repository[1] that was updated on a weekly basis. This repository contained learning materials and including programming exercises for the students to complete. Exercises solutions could then be pushed back to GitHub where student could receive feedback. As well as this main module repository assessments were also managed via GitHub to ensure consistency with the provision and submission of module work.

1 A repository is a projects main folder, which contains all the files and keeps track of the files history. Students could download these repositories and their work then push the new changes back to GitHub.

To aid introduction a two-hour class session was dedicated to teaching students about Git[2] and GitHub. In these sessions' students were given demonstrations on its usage, including guided walkthroughs on accessing and pushing work back to GitHub. Students were encouraged to use GitHub Desktop[3] as it simplifies repositories management in comparison to other more technical methods. PDF Guides, videos and links to additional resources were also made available to make it easy for student to gain help if required.

From the teacher perspective implementation was managed with the use of GitHub Classroom[4], a tool that automatically creates required repositories for each student. Additional shell scripts were also used to streamline download and access student repositories. Providing feedback was factored into the weekly workload with time spent on Fridays reviewing work submitted to student repositories. Additional feedback was provided to CL1 students on five assessed in class tests, which was all managed through GitHub.

Observation & Findings

The use of GitHub was overall positive, but not without issue. Most importantly student's response was positive, measured through Likert scale questions in end of module feedback (1 strongly disagree, 5 strongly agree). On average students found GitHub useful (3.9), found feedback useful to improving work (3.9), would continue using it (3.7) and saw it as valuable to their career (3.8).

"the feedback i received helped me to understand where i was going wrong"

Whilst this points to positive experiences there were also issues with the implementation. The way student logins are handled on the University PC's meant multiple users were not handled well by GitHub Desktop. This led to students receiving error messages that required workarounds that altered a usually straight forward workflow. This led to some students perceiving GitHub as a poor tool and avoiding its use. These issues were felt by more by CL2 students with the earlier scores in relation to GitHub's value going up when looking at CL1 alone. CL2 students also showed higher agreement with finding GitHub difficult to use (3.55) and would like more tutorials on its use (3.36) than CL1 students (2.8 / 2.8).

"Github is complicated and unnecessary"

These differences may be due to GitHub altering the workflow CL2 students were used to from the previous year. Whilst for CL1 as new students it is '*the way of working'* in the module. CL1 students also got greater exposure due required usage for submitting five in class tests. Differences in adoption were also seen within cohorts. This might be due to an additional burden on an already difficult to master subject, and struggling students wanting to avoid detection by not pushing work

2 Git is the version control system that GitHub integrates with. GitHub provides remote storage and other social features for Git based repositories

3 https://desktop.github.com/

4 https://classroom.github.com/classrooms

back to GitHub for review. Thus, greater emphasis may be required to encourage students to push work to GitHub, especially as those who committed[5] their work more often tended to appear in higher grade boundaries[6].

Recommendations

Based on the above it is important that time is given to adequately introduce new technologies. Not only providing guided demonstrations but also making clear the expectations of its use. Whilst demonstrations were given some students were observed to still struggle, in part due to the technical issues described above, but also a potential assumption of knowledge. When teaching computing students and students deemed '*digital natives'*, there can be assumption that they will adapt easily and already have a level of knowledge. However, whilst they may be confident with technology in their personal lives, this does not extend to their learning life (Newman and Beetham, 2017). As the implementer of the teaching innovation there is an inherent higher-level understanding. Thus, taking a step back and adopting a beginner's mindset is required when introducing this to students. The provision of learning resources beyond the class demonstrations also becomes valuable for students.

Conducting pilots as close to the real usage scenario as possible is also important, as per the ADDIE Model conducting these prior to implementation (Branch, 2009). In preparation for the GitHub implementation tests were conducted on the desired workflow. However, these were only ever conducted on staff logins and the issues described above only arose during usage with students in the classroom setting.

Finally, if the technology innovation adds to existing workload ways to streamline the process are important. Tools that eased the workflow in this project were invaluable to maintain the added commitment of providing feedback outside of class. Consistent weekly feedback cycles were also important. However, it was observed that students also need to be aware of such cycles, with some questioning why they had not yet received feedback. Thus, managing expectations both in terms of what you expect of the students from implementing the technology and what they can expect from you also important.

5 A commit is the terminology to describe saving changes to the repository.
6 Based on comparing number of repository commits to overall grade for CL1 students

References

Branch, R. M. (2009). *Instructional design. The ADDIE approach.* New York: Springer

Jonsson, A. (2013) 'Facilitating productive use of feedback in higher education', *Active Learning in Higher Education*, 14(1), pp. 63–76.

Lahtinen, E., Mutka, K. A. & Jarvinen, H. M. (2005). A study of the difficulties of novice programmers. In *Proceedings of the 10th Annual SIGSCE Conference on Innovation and Technology in Computer Science Education*, Monte da Caparica, Portugal, June 27–29, 2005 (pp. 14–18). New York: ACM Press.

Newman, T., & Beetham, H. (2017) *Student digital experience tracker 2017: the voice of 22,000 UK learners*. JISC. Available at: http://repository.jisc.ac.uk/6662/1/Jiscdigitalstudenttracker2017.pdf [Accessed April 30 2019]

Nicol, D., & Macfarlane, D. (2006). Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice. *Studies in Higher Education*. 31(2). pp. 199-218.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(1). pp. 137-172.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school: A Pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73-88

Wolsey, T. (2008). Efficacy of instructor feedback on written work in an online program. *International Journal on ELearning*, 7(2), 311–329.

Resources

Github - https://github.com/

Github Desktop - https://desktop.github.com/

Github Classroom - https://classroom.github.com

Github Classroom for Teachers (Guides and Scripts for repository management)- https://github.com/jfiksel/github-classroom-for-teachers