LEEDS INSTITUTE *for* TEACHING EXCELLENCE

Development of an automated assessment and feedback platform

Dr Craig A. Evans and Dr Sam Wilson

Project Report

Feb 2020

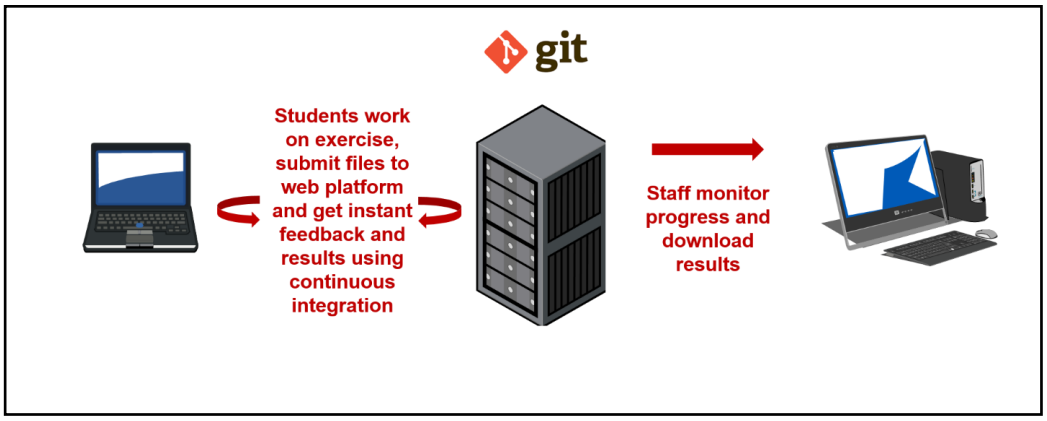
UNIVERSITY OF LEEDS

Project Overview

When teaching a practical, skills-based subject such as computer programming, it is important to provide plenty of opportunities for students to practice and develop their coding skills. Students typically attend laboratory sessions and complete a series of learning exercises, in which they can ask for assistance and staff are able to provide verbal feedback. However, the nature of developing a skill means that students must also devote their private study time outside of the classroom to further practicing and honing their skills. Naturally, this setting makes it harder for staff to provide assistance and feedback.

Students are often required to submit their solutions to learning exercises in order to receive formative feedback. However, the growth in student numbers in recent years and the sheer number of practice exercises completed means it is unfeasible for staff to manually provide this much feedback in a timely (and hence useful) manner.

Our project aimed to develop an online platform to which students can submit their solutions to these exercises and receive automatically generated, instant and personalised feedback.



By automating the process, we can achieve a step-change in the amount of feedback students receive.

The platform was also designed to be used for summative assessment, thus freeing up marking time.

Project Objectives

- Develop a secure online platform through which students can submit solutions to learning exercises
- Develop a series of testing scripts that can automatically test submissions and provide useful feedback to students
- Develop a series of test scripts that can be used for marking summative assignments, collating results for staff and returning feedback to students.

Methodology

The online platform employs industry-standard software engineering practices such as version control, unit testing and continuous integration (CI). Staff are able to create instances of modules and enrol students onto them. Various items of coursework or learning exercises

can be added to the modules. Students submit solutions via the web front-end, automatically triggering testing scripts that mark submissions and generate personalised feedback.

The testing scripts are typically written in Python and use various software tools to analyse the submissions. As well as providing feedback on the functionality (i.e. does the code do what it is supposed to do?) the programme also offers suggestions of possible causes of problems, highlights possible bugs and provides contextual information on the code quality such as the code style and commenting.

Outcomes

A series of test scripts have been produced. These test scripts are able to grade and provide feedback for a range of programming languages (C/C++ and Python) and simulation packages (Logisim and Nand2Tetris). The test scripts were used to grade summative and formative assignments in semester one of the 2019/20 academic year in several modules across the School of Electronic & Electrical Engineering, School of Computing, School of Mathematics and the Joint Engineering School at South-West Jiatong University in China.

We estimate that around 4500 assignments have been automatically graded up to January 2020. The wide range of assignment types means that it is not possible to produce a completely accurate figure, but assuming that each assignment would take 10 minutes to manually grade, this equates to 750 hours of staff time (or around 19 working-weeks). This type of marking is often done by teams of Postgraduate Research students in order to reduce the turnaround time. Based on usual payment rates, this equates to a cost-saving of around £9000.

It is also worth pointing out that when marking this number of assignments, it is inevitable that mistakes will be made by human markers. To date, we are not aware of any instances of incorrect grading made by the platform.

Challenges

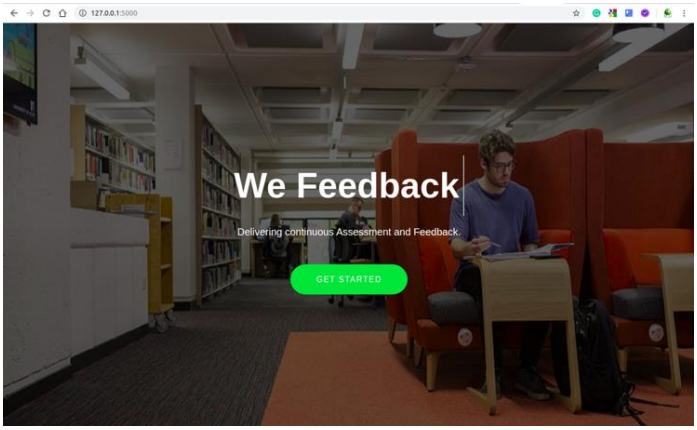
A large challenge in the project was providing useful and personalised feedback. Automated code testing is common practice in industry. However, these tests are often on a PASS/FAIL basis and rely on the developer to determine why a test failed. This type of PASS/FAIL feedback is not useful to novice programmers, who often lack the knowledge and experience to find and fix the problem. Therefore, we developed the platform so that as well as being able to detect that the solution was incorrect, it could analyse the returned values and suggest possible errors made in the code that would cause those values to be returned.

Another challenge was developing the test scripts to be robust enough to handle ‘live’ code developed by students. The reality of learning to program means that code produced by novices can contain an unthinkable number and range of problems. For example, returning the wrong type of data, not returning data, or getting stuck in an infinite loop. All of these can cause the testing scripts to hang.

A final challenge was security; ensuring that data is safe and no personal information is collected.

Next Steps

The next step of the project is to integrate the testing scripts with the online platform. This will allow students to submit their solutions to the learning exercises at any time and receive useful, personalised feedback.



Once this step is complete, the platform will be collecting a large amount of data, such as the number of submissions made. This will be extremely useful for staff and allow a learner analytic approach to be taken. For example, we will be able to identify areas of the curriculum that students find more difficult, as well as identify underachieving students and those at risk of failing.

Furthermore, we are looking for staff from the University (and beyond) to adopt the platform into their own teaching. A Share, Adapt, Adopt workshop will be run in LITE during 2020.