

Assessment Case Study

Feedback in Computer Science – Meaningful, Consistent, On-time, Extensive, Popular (and Automated!)

Practitioners: Matthew England, ab9797@coventry.ac.uk and David Croft, ac0745@coventry.ac.uk, School of Computing, Electronics and Mathematics, Faculty of Engineering, Environment and Computing

Course: BSc Computer Science

Student Level: Undergraduate (Level 4)



AN EXAMPLE OF ASSESSMENT THAT INVOLVES...

- Formative assessment
- Automated feedback

AIMS

The **Programming and Algorithms** (4000CEM) module is taken by students with huge diversity in study skills, prior programming experience, and interests. The module is taken by **hundreds of students** each year, but the majority of actual contact time is delivered in groups of 40 or less, as it needs to take place in **PC labs**. This necessitates a **large module team** and thus adds the issue of **consistency** among the labs and teachers.

Against this backdrop, the module team aimed to vastly increase the **amount of formative feedback** students receive; whilst increasing its **availability** (as it does not have to take place in labs) and making it more **consistent** while still **meaningful**.

ACTIONS

Students were provided with licences for a cloud computing environment ("Codio"). This enables them to be presented with tasks and then to undergo a cycle of:

- (re)attempt task; - run automated tests; - read feedback

This is similar to the development and testing environment used in **professional practice**, and can be undertaken at any time or location. Although feedback from Codio is **automated**, it is also:

- **Meaningful:** it points out what has gone wrong *for that student in that attempt* at the task.
- **Consistent:** all students get the same opportunity to access feedback of the same quality.
- **On-time:** students can access it whenever they want.
- **Extensive:** there are multiple tasks every week each with feedback.
- **Popular:** See Impact section below.

Timetabled lab sessions were used for staff to **support students** who were unable to end the cycle, and **discussion** (e.g. what lessons should be drawn from the tasks and links to theory).

Staff were thus able to spend **more time engaging in deeper learning**, and less pointing out common bugs. Staff subsequently started using this feedback mechanism for (part of the) **summative assessment**.

IMPACT

- The module's most recent MEQ had:
 - * **93%** satisfaction overall, and
 - * **90%** satisfaction with feedback.



- This data indicates that **students do accept automated feedback**; in fact, they seem more willing to accept it than human generated feedback delivered informally (talking in the lab).
- In recent years there has been a steady increase in **overall satisfaction, feedback satisfaction** and **student performance** in line with the introduction of this initiative.
- Anecdotally teaching staff have reported **better attendance** on this module than many others in the course.
- The module team have **produced research** based on this intervention (see Further Reading below).

TOP TIPS

- There is scope to use **automated feedback in other modules** where the right answer can be judged automatically (e.g. programming, mathematics, languages).
- Automation is **not a replacement for human interaction**. The module team retain the same amount of contact hours. Rather, the automation enables the human interaction to be more focused on covering deeper, more important topics.
- Automation does **not save staff marking time** in this module. Students still undertake group projects which are marked by group presentation and viva.
- The formative tasks and tests took considerable time to develop, but once created they can be **used for several years** with only minor maintenance.
- Codio was the tool used in this case, although the development of tasks and tests was undertaken by the module team. **Moodle's Coderunner** feature provides similar functionality.

FURTHER READING

Croft, D. and England, M. (2019) 'Computing with Codio at Coventry University.' in *Proc. 3rd Conference on Computing Education Practice* (CEP '19), Article Number 16, ACM <https://doi.org/10.1145/3294016.3294018>