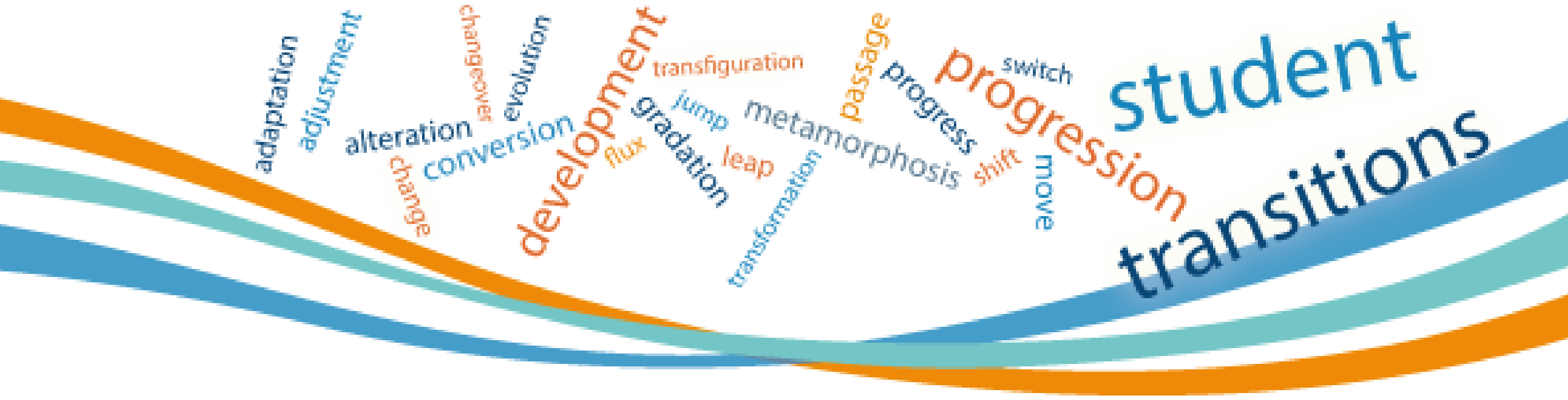

Student Transitions: Example of transitions practice

Title: Helping 1st Year CS Students to Become Independent Learners Through Automated Feedback

Transition(s) the practice supports: the aim of this is to identify where this would sit on the planned transitions map.
Type 2: New proposals that address the theme of Student Transitions

This proposal addresses the transition students have to go through at their first year at uni: from being school pupil to being an independent learner.

Abstract: a brief outline of the practice – this will allow all your institutional examples to appear on the one page of the website, linking to further information. This should be no longer than 120 words.

Learning how to programme computers is one of the major hurdles for our first year Computer Sciences (CS) students across Heriot-Watt campuses. A multi-national, multi-institutional assessment of programming skills of first-year CS students found that many students do not know how to program at the conclusion of their introductory courses (McCracken et al., 2001). Some of the potential issues for this poor performance discussed in the study are constrained lab time, large classes and insufficient feedback. In previous work, we found that automated feedback can help students to become independent learners when faced with new programming tasks (Rieser, PGCAP report, 2014). Here, we propose to extend this previous work to help first year CS students to become independent learners and master the art of programming during their first year at the university.

Description: (this is the information to which the abstract would link) a more detailed description of the practice with an indication of what led to its development, how long it has been in place, and why it benefits students

In particular, we propose to develop a platform based on JUnit testing, which is an automated test harness for Java programming to generate formative feedback to the students. This type of automated testing has been shown to be successful in, for example, teaching students how to code in HTML (Vihavainen et al. 2013).

JUnit tests are a piece of Java code, which allows to automatically find errors in the code and detect missing aspects. This automatic analysis then can be used to generate meaning full feedback, for example, in terms of clues and tips how to improve the code.

This platform will be implemented to be part of a newly developed shared coursework between Software Development 2 and 3. Both courses run in parallel in the second semester. That is, before attending these courses, students will already have had some basic introduction to programming in Java in Software Development 1, and are now expected to deepen their knowledge and skills in these two advanced courses.

Bibliography:

Michael McCracken, Vicki Almstrum, et al. 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In Working group reports from ITiCSE on Innovation and technology in computer science education. ACM, New York, NY, USA, 125-180.

Arto Vihavainen, Thomas Vikberg, Matti Luukkainen, and Martin Pärtel. 2013. Scaffolding students' learning using test my code. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13). ACM, New York, NY, USA, 117-122.

Contact details: name and email address of key person who can discuss the case study in more detail. (with their permission agreed before submission)

Dr. Verena Rieser
v.t.rieser@hw.ac.uk
http://www.macs.hw.ac.uk/~vtr1/