Description

The postgraduate module Advance Software Engineering adopts a research-oriented learning approach (Blackmore and Cousin 2003; Brew and Prosser 2003; Healey and Jenkins 2009; Healey and Jenkins 2009) to expose students to software engineering research. The aim is to provide students with opportunities to discuss and evaluate the latest published papers in the field and get to critically evaluate state-of-the-art software engineering literature.

In practice, students are exposed to a rigorous analysis of claims and counter-claims made by published software engineering researchers and develop a systematic approach to careful evaluation of published work. It also develops familiarity with a broad range of current scholarship in software engineering work. A routine element of peer-feedback runs through the module.

Approach

The module takes the form of a seminar series in which students do the main presentation after a short introductory tutor presentation. Each session bears on a specific theme. The number of presentations obviously depends on the number of student enrolled on the module - and is therefore not directly transferable to large groups, unless group presentations are introduced. The student presenting the paper has to hand-in a report in which they assess the article. The other students write a short abstract of one of the presentations given during the session.

The tutor gives feedback in the following ways:

- Generic class feedback. This was very important at the start of the module especially so that all students get used to the criteria for a critical and reflective presentation. Feedback on issues common to all students is given at the end of each session. A summary note is posted on Blackboard for all.

- Specific individual feedback is provided on a bespoke School of Computing assignment feedback sheet. In the first 2 deliveries, the submission was on Blackboard and the sheet was uploaded for each student every week. Although it was time consuming, it was still manageable for small cohort. With a large cohort, it started to be unmanageable. We therefore adopted rubric. In designing the rubric(s), we not only identified the elements of the assignment that would be evaluated, but we also identify the range of performance under each element. The rubric presented to students detailed criteria for assessing work.
Outcomes

- Students examine in depth some of the techniques and tools in Software engineering via published research.
- Students develop their ability to write briefs, and coherent abstracts on a presentation they have attended; this helps note-taking and focusing during the presentation.
- Students become more critical when evaluating and discussing published work.
- Students who present need to read in depth, and critically evaluate, a recent paper in their subject specialism. This prepares students towards writing for publication.

Impact

This research-led approach has promoted good attendance. It has improved the student experience as they are actively engaged with the topic (as critique or as reporter of the presentation). It promotes good interaction in the class especially when it is a small cohort. This module obtained excellent results (over 90% pass rate). Through word of mouth, this module also became quite popular with subsequent cohorts.

Lessons learned

**Do:** Prepare students; set clear instructions and criteria for presentations and feedback; support students prior to presenting; allow for constructive criticism.

**Don't:** Leave students to their own device; allow strong voices to take over the discussions; leave it too long to provide feedback.

**What makes this approach particularly successful?**

Students are actively engaged and focused on critically evaluating state-of-the-art research in their field.

Where next?

This approach could be adapted for large groups and/or for undergraduate students. It could also be used as a framework for problem-based teaching, where a problem is examined every week. An analysis of the problem and a solution is discussed in seminar.

References

