Quality Assurance and Testing

- Dr Colin Higgins.
- Room C10b, CS, Jubilee.
- cah@cs.nott.ac.uk
- [Link](http://www.cs.nott.ac.uk/~cah/G53QAT/G53QAT.html)
- Slides based loosely on Galin.

Lecture 1 – Introduction
The Software Quality Challenge

- Overview of the Module
- The uniqueness of software quality assurance
- The environments for which SQA methods are developed

**Learning outcomes:** Identify unique characteristics of software as a product and as production process that justify separate treatment of its quality issues.
Recognise the characteristics of the environment where prof. s/w dev. and maintenance take place.
Explain the difficulties faced by s/w development and maintenance teams as a result of this environment.

Teaching Activities

- Tuesday 16-17 hrs in A25-JBCSOUTH, Thursday 13-14 hrs in B43-JBDEARING
- Notes for the lectures are available from the following url: [Link](http://www.cs.nott.ac.uk/~cah/G53QAT/G53QAT.html)
- Students are also expected to take notes in class
- Exercises and extra notes will be available for some lectures

Assessment

- Written examination (50%).
- Coursework (50%). Building a small web based portfolio on a given topic, giving a presentation on your topic. Marking other peoples topics and presentations.
- This depends on numbers etc. Details will be given in the module web page later.
The uniqueness of the software development process

- Compare s/w with industrial products
- High complexity
- Invisibility of the product
- Limited opportunities to detect defects ("bugs")

SQA Environments

- Students develop s/w as part of their education.
- S/w amateurs develop s/w as a hobby.
- Professionals in eng, economics, management etc develop s/w to assist in their work.
- S/w professionals (systems analysts and programmers) develop s/w (& firmware) products as prof career objective in teams in a variety of organisations (s/w house, s/w dept etc)

The characteristics of the SQA environment process

- Being contracted
- Subjection to customer-supplier relationship
- Requirement for teamwork
- Need for cooperation and coordination with other development teams
- Need for interfaces with other software systems
- Need to continue carrying out a project while the team changes
- Need to continue maintaining the software system for years

Cooperation and coordination scheme for a software development project team
Salary software system - an example of software interfaces

- Attendance control system
  - Input interface
    - Monthly attendance report, including overtime calculations
- Salary processing system
  - Output interface
    - Money transfers to employees’ bank account accounts
- Bank information systems