## Subject Knowledge

### Aims
The module approaches the issue of quality assurance in the software development process at an advanced level. This includes a rigorous account of the strategies for software testing and quality control, and the introduction of software metrics for quality assurance and project cost estimation. The module is focussed around the notion of software process improvement.

### Learning Outcomes
Students will be able to describe how quality issues affect each aspect of the software development life-cycle. They will be able to choose appropriate strategies for software testing and validation, and discuss how to implement them. They will be able to demonstrate understanding of the theory of software metrics and be able to make software measurements in practice. They will be able to relate quality to the current standards for process improvement.

### Methods
Class sessions together with course notes; recommended textbooks; worksheets; additional hand-outs including articles, case studies and research papers; web resources.

### Assessment
Marked coursework, group presentation, written examination.

## Skills

### Aims
Students will learn how to research current issues in software quality assurance, and how to present their findings. They will learn how to turn theoretical ideas into practical process improvement steps in an industrial context.

### Learning Outcomes
Students will be able to research a given topic using a variety of sources including books, current articles and research papers and web-resources. They will be able to give a written account of their findings (suitable for inclusion in a company report). They will be able to give a seminar-style presentation of their findings using appropriate audio-visual aids.

### Methods
Class sessions together with worksheets. Library and web resources.

### Assessment
Marked coursework, group presentation, written examination.

## Explanation of Prerequisites
The module will build on many of the ideas that are introduced in C01006 and CO2006 so these modules are essential. The sections on testing will also require experience of programming in Java. Software Engineering is as much a practical subject as an intellectual discipline so it is an advantage to have had the experience of working in a team on the CO2015 Software Engineering Project.

## Course Description
A major aim of Software Engineering is to ensure the quality of the final product of the software development process. Quality is not an extra that can be added at the end of the process. To achieve it one must consider how each stage contributes to the quality of the final product. Quality assurance should be seen as an intrinsic part of the software life-cycle. One key mechanism of quality control is software testing, another is inspections and reviews. However, these should be implemented as part of a wider Quality Assurance Plan.
The first step towards quality is to understand what it is and how to measure it. The overall quality of a product is a rather vague idea that cannot be measured directly. It can be seen as an amalgamation of different attributes: correctness, reliability, maintainability, ease of use, and so on; which can be measured by developing the right software metrics. The use of metrics is thus an important tool in quality assurance. Furthermore, quality cannot be considered without reference to the associated cost. Metrics can also be used to gauge the size and complexity of software and hence are employed in project cost estimation.

This module will look in depth at the issues of software quality assurance from an industrial perspective, exploring the techniques available and how these might be employed. It will focus around the idea of software process improvement, as seen in the SEI process Capability Maturity Model and SPICE (ISO/IEC 15504).

**Detailed Syllabus**


**Inspection** Code walk-through, inspections, reviews. Comparison of different approaches. Effective follow-up: collection of data, review feedback.

**Testing** Different levels: unit testing, integration testing, system testing, validation. Structural testing: coverage techniques. Behavioural testing: domain testing, finite state testing. Mutation and fault seeding. Tools and instrumentation.


**Reading List**


**Resources** Course notes, web page, study guide, worksheets, handouts, lecture rooms with OHP and data-projector, past examination papers.

**Module Evaluation** Course questionnaires, course review.