
CO7095 Computer Systems

Credits: 15 **Convenor:** Dr N. Walkinshaw **Semester:** 1st

Prerequisites: *Essential: Knowledge of software requirements and design procedures, knowledge of Object-Oriented programming concepts*

Assessment: *Coursework, two class tests: 40% Three hour exam in January: 60%*

Lectures: 32 hours

Surgeries: 10 hours

Laboratories: 10 hours

Private Study: 90 hours

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; research papers; web resources.

Assessment Marked coursework, 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).

Methods Class sessions together with work sheets.

Assessment Marked coursework, written examination.

Explanation of Prerequisites Software quality is a broad concept, that encompasses several aspects of software development. On the one hand, software quality is concerned with the complexity of the software source code and its design. To learn about this, it is essential to have a prior understanding of Java and Object-Oriented design (CO1003, CO1005). However, there is also a broader picture; quality depends on the ability of developers to properly plan development, to estimate costs, and to follow suitable procedures to ensure that quality is maintained. Accordingly, the student will ideally have studied preliminary modules on requirements engineering (CO1008) and project management and professionalism (CO2012).

Course Description The course will introduce various concepts associated with quality and quality assurance. It will show how these do not only revolve around software as an entity, but also depend upon the underlying development procedures, and the ability to improve upon these as the software and its stakeholders evolve over time.

Detailed Syllabus

Quality Defining “Quality” with Software Quality Models. Different (often competing) perspectives of software quality.

Risk analysis and management Measurement Theory of measurement. Project size/cost estimation. Quality metrics, cost metrics and process metrics. Statistics: data collection and analysis.

Software inspections and testing An overview of approaches to assess the quality of a software system and its design. These include human-centric methods such as software inspections / code reviews, as well as software testing approaches. For the latter we will look at both white-box coverage-based testing approaches, as well as “black-box” functional testing techniques.

Tools and instrumentation Process improvement frameworks such as the Capability Maturity Model. Software quality assurance models and standards.

Reading List

[A] S. Kan, *Metrics and Models in Software Quality Engineering*, Addison Wesley.

[B] N. E. Fenton, *Software Metrics: A Rigorous Approach*, Chapman & Hall.

[B] I. Sommerville, *Software Engineering*, Addison Wesley.

Resources Course notes, Blackboard page, study guide, worksheets, lecture rooms with data-projector, past examination papers.

Module Evaluation Course questionnaires, course review.