MODULE **DESCRIPTOR**

MODULE TITLE	Software Development		
MODULE CODE	CO2401 (L5)	**CREDIT VALUE**	20 credits / 10 ECTS
SCHOOL	SCHOOL OF SCIENCE		

MODULE **AIMS**

1. To develop communication skills required for effective software development
2. To explore approaches to identifying software requirements and developing software
3. To emphasise the importance of HCI in the software development process
4. To develop a systematic approach to software quality
5. To enable the students to compare software development approaches

MODULE **CONTENT**

Quality

- Quality criteria, e.g. meeting user needs, delivery on time, robustness, maintainability
- Safety Critical Systems: hazards, risks and fault tree analysis
- Process and product quality
- Standards & methodologies
- Testing: testing strategy, test case design, test harnesses, mocking/stubbing
- Usability evaluation
- Reviews, inspections

Software requirements

- Agile and traditional approaches to identifying and recording requirements

User interface design

- User needs analysis
- Colour, Font, Navigation, Affordance of controls
- User error recovery

Software design and development techniques and tools

- Agile and traditional techniques for software development (e.g. TDD, CRC cards, modelling using UML - to include use-cases, class, activity and sequence diagrams)
- Object-Oriented analysis and design
- Software tools to support software development (e.g. diagramming, TDD, mocking/stubbing, test coverage, static analysis, bug-tracking, documentation generation, configuration management and change control)
- Software Reuse

INTENDED **LEARNING OUTCOMES**

On successful completion of this module a student will be able to:

1. Elicit software requirements and design an appropriate software solution
2. Design and evaluate a user interface using established HCI principles
3. Evaluate approaches to ensuring software quality
4. Choose and apply appropriate software development approaches, methods, and tools for a given problem

TEACHING METHODS

A combination of lectures with tutorial / practical sessions.
The Lectures will cover the theoretical underpinning, while tutorials will include role-playing, group discussions, critical evaluation of prototypes and design exercises.

The requirements elicitation stage may be acted out through role-play of a customer and development team. Students develop a set of interface screens based on the customer requirements and underpinned by published HCI research. Students will then present the screens to the customer in another role-playing exercise and feedback from the customer will be used to improve the interface. Through further role-play, the screens will be evaluated by "experts" and by "users" using published usability testing methods.

Practical sessions will enable the students to develop designs and implement prototypes and receive formative feedback from both the tutor and through role-play exercises from the "user".

Practical work will allow students to explore tools to support software development techniques

The assessment will comprise the development of a software solution to a given problem which illustrates the full development lifecycle. Groupwork may be used to allow students to appreciate the benefit of design documents to support communication and thought and to appreciate the benefit of "inspections".

ASSESSMENT METHODS

This module is assessed through an examination (50%) and a coursework (50%).