MODULE CODE:	**CHS2517**
MODULE TITLE:	**LARGE SYSTEMS ENVIRONMENTS**
SCHOOL/S INVOLVED IN DELIVERY:	Computing & Engineering
NAME OF COURSE(S):	MEng, BSc CSD, BSc SD with …
MODULE LEADER:	DR C D Newman (Informatics)
LOCATION:	Queensgate
MODULE TYPE:	Core on MEng SE, BSc SD, BSc Comp, BSc S&FC
CREDIT RATING:	20
LEVEL:	H
LEARNING METHODS:	Supervised learning: 44 Hrs Unsupervised Learning: 156 hrs
PRE-REQUISITE/S:	None
RECOMMENDED PRIOR STUDY:	None
CO-REQUISITE/S:	None
PROFESSIONAL BODY REQUIREMENTS:	N/A
GRADED OR NON-GRADED:	Graded
BARRED COMBINATIONS:	None

SYNOPSIS:

This module provides an analysis of the problems associated with large-scale software development projects and the management practises, procedure, and tools designed to address these issues. Such tools will vary from simple stand-alone tools to integrated Software Engineering Environments (SEE's). Mechanisms for determining the quality both of the software systems and of the design process itself will also be addressed.

OUTLINE SYLLABUS:

The Software Crisis and the Software Engineering Lifecycle.
The problems inherent in large-scale software projects and the need for Project Management, including CM and VC.
Project Costing, Estimating, and Planning and methods to support these.
Software Tools, Computer Aided Software Engineering (CASE), Integrated Project Support Environments (IPSE's), Software Engineering Environments (SEE's).
Defining and measuring software quality - Software Metrics, Software Quality Models, Process Maturity Model.
Risk analysis, assessment, management and mitigation using tools, techniques, classifications and frameworks.

LEARNING OUTCOMES:

Knowledge and Understanding Outcomes
Upon completion of this module, the student will be able to:

1.1 Discuss the problems inherent in large scale software development projects and the need for software project management including configuration management (CM) and version control (VC).
1.2 Have an appreciation of where risks occur and how they might be address.
1.3 Describe the uses and limitations of current methods of software project management, CM and VC.

1.4 Discuss the need for and application of software tools which support software project management, VC, and CM, from individual stand alone tools through to full-scale SEE's.
1.5 Discuss and evaluate techniques for assessing the quality of software artefacts and the software process itself.

Ability Outcomes
Upon completion of this module, the student will be able to:

2.1 Make use of current methods and tools applicable to the management and completion of large scale software development projects.
2.2 Produce risk control documents such as a risk impact assessment table, risk register and risk information sheet etc.
2.3 Take measurements and make judgements about the quality of software systems and their development.
2.4 Comment critically on current management practises, methods and tools.

ASSESSMENT STRATEGY:

Formative Assessment

The formative assessment will be by providing comments on a selection of the tutorial exercises submitted for assessment once the students have had time to enhance their tutorial work by the individual student's own self-study efforts. The students will be encouraged to discuss the returned tutorial work during subsequent tutorial sessions. The tutorial exercise chosen for formative feedback will contain both a practical element and a reflective critique and will cover all Learning Outcomes . The formative assessment will prepare the students for the summative assessment.

Summative Assessment

The summative assessment will be via an end of year examination. The examination is not eligible for tutor reassessment.

Assessment Tasks

The examination will be of two hours duration and will cover all Learning Outcomes. The examination will also cover material that the students' will have read during their self-study efforts. The examination paper will contain five questions and candidates will be asked to attempt any three. The examination is the only and final piece of assessment and will be marked anonymously.

The formative tutorial work will be assessed on the following criteria:
1) quality of the discussion, the completeness of coverage of the material, and the relevance of the discussion to the problem.
2) appropriate range and depth of the discussion the selection and discussion of the components or topic chosen for inclusion in the work submitted.
3) the presentation of the material, including the readability of the work.

Assessment Criteria
The examination will be marked in accordance with normal University practice and the answer scheme produced by the module leader.

LEARNING STRATEGY:

The weekly sessions will be used to deliver the background material and to demonstrate the application of the methods and tools addressed within the module. This will be supported to allow the concepts and techniques to be put into practice through the use of exercises and case studies.

At three points in the course, for example Reading Week in the first term, week twelve and Reading Week in term two, students will be given tutorial sheets for formative assessment. This will allow feedback to be provided before the end of term one, at the beginning and end of term two. The questions/case studies chosen for formative assessment will be constructed to facilitate confidence to undertake and excel in the written examination in the summer.

APPENDIX

INDICATIVE REFERENCES:

Pressman, R S Software Engineering: A Practitioners Approach (6th Edition, European Adaptation) McGraw Hill, 2005

Sommerville, I Software Engineering (9th Edition) Addison-Wesley, 2011

Avison, David & Fitzgerald, Guy. Information Systems Development (methodologies, techniques & tools) McGraw Hill 4E 2006. {3E 2002}

van Vliet, Hans. Software Engineering – Principles & Practice, Wiley 3E 2008

Gardiner, P D, Project Management (A strategic planning Approach) Palgrave MacMillan 2005

Marchewka, J T, Information Technology Project Management, 2E Wiley 2006

Kruchen, P. The Rational Unified Process An Introduction Pearson 3E 2004

Kan, Stephen H, Metrics & Models in Software Quality Engineering,Addison-Wesley 2E 2003

Braude, Eric J., Bernstein, Michael E. Software Engineering Modern Approaches (Second Edition) John Wiley & Sons Inc. 2011