

LEARNING AND TEACHING COMMITTEE

Module Specification

Please complete all sections of the form, referring to the guidance notes where necessary.

| | | | |
|---|---|--------------|----|
| School and Subject Group | School of Engineering and Applied Science/ Computer Science | | |
| Module Code | CS4670 | | |
| Module Title | Software Process Management | | |
| Module Type | Taught module | | |
| Module Replaces (where appropriate) | | | |
| Date of introduction of new module | October 2008 | | |
| Level | 7 | Credit Value | 20 |
| Programme(s) in which module is available | MSc IT Project Management, MSc Software Engineering | | |
| Involvement of Other Schools | ABS (for MSc IT Project Management) | | |

| | | |
|---|----------------------------------|-------------|
| Resource Split | | None |
| Name of Module Co-ordinator | | Dr Hai Wang |
| Related Modules | Pre-requisites | |
| | Co-requisites | |
| | Prohibited Combinations | |
| | Minimum and Maximum Intake Sizes | 2 40 |
| <p>Aims of the Module</p> <p>A software process is a complete system of work through which to advance from identification of need to an operational software end-product. The module will:</p> <ul style="list-style-type: none"> • Develop a systematic understanding of a software process appropriate to an object-oriented context, together with an appreciation of alternative approaches. • Develop an in-depth understanding of how software metrics can support and improve the management of projects, both tactically and at the more strategic level of an organisation's software process. • Enable students to acquire associated practical software engineering skills. | | |

Summary of Content

Concept of software process. Overview and detailed exposition of the Unified Process, largely through suitable case studies.
Agile software development. Some of the well-known agile software development methods, e.g. Extreme Programming.
Software development for reliability and reuse: design and programming by contract; role of assertions and formal notations;
Concept of design patterns, including outline and application of a representative range of examples.
Alternative methods, e.g. Component-Based Software Engineering.
Software engineering: management aims; software process; tactical & strategic issues
Software metrics foundations: principles and techniques of: measurement, estimation, experimentation, data analysis
Software project management: planning, monitoring & control, including earned value analysis; productivity; staff skillsets
Risk management: risk analysis, assessment, control, resolution
Software configuration management: configuration control, release management, configuration process
Software cost management: software cost estimation, software, COCOMO model
Quality management: nature of quality; quality control, assurance and management systems; quality models and metrics; software management – testing, version control, and integration. Process metrication: GQM (Goal-Question-Metric) paradigm.

Summary of Methods and Frequency of Teaching

Teaching methods include:

- Lectures: Two-hour lectures per week for TP1 and one hour lecture per week for TP2. These are used to present theoretical and practical aspects of software process management technologies.
- Labs: One-hour lab every two weeks for TP1 and TP2. These are used to allow students to practice the technologies presented in the lectures.
- Tutorial: One two-hour tutorial every two weeks for TP1 and TP2. These are used to allow students to practice, present and discuss the related technologies.

Summary of Methods of Assessment and Feedback, including Formative Feedback

Three hour written examination (60%), Practical assignment (20%) and Class tests (20%).

- Assignments will be marked and returned within three weeks after the submission.
- Student presentations and discussions will take place during lectures and tutorials.
- There are further resources on the course web site.

| Module Outcomes – what the student should gain from successful completion of the module: | Learning and Teaching and Assessment Strategies to enable outcomes to be achieved and demonstrated | |
|--|--|---|
| | Learning and Teaching Methods | Assessment Methods |
| <p>Knowledge and Understanding</p> <p>The structure and application of a software process. Relevant tools and techniques, and their application in example problems. Principles of, and techniques underpinning, software metrics, and their use in improving an organisation's software process and project management</p> | Lectures, tutorials and appropriate reading | Exam including problem-based questions |
| <p>Intellectual Skills</p> <p>Ability to apply a disciplined software process to work systematically towards a complete software solution to a specified problem. Ability to select/specify appropriate software process metrics. Capability with a wide range of appropriate quantitative-based techniques.</p> | Lectures and tutorials, supported by appropriate reading. | Exam and coursework |
| <p>Professional/Subject Specific Skills</p> <p>Enhanced ability in applying:</p> <ul style="list-style-type: none"> • Suitable software development methods. • A suitable programming language (currently Java). • Ability to apply measurement and estimation techniques in software project management. | Lectures, tutorials and practical work, appropriate professional literature | Coursework and problem-based exam questions |

| | | |
|--|---|---|
| <p>Transferable Skills</p> <p>Effective group interaction capability. Ability to assess the relevance and potential benefit of metrics-based techniques to projects in areas other than software engineering.</p> | <p>Lectures, tutorials and practical Work</p> | <p>Exam and coursework</p> |
| <p>Please provide either or both of:</p> | | |
| <p>(i) Introductory Learning Resources</p> | | <ul style="list-style-type: none"> • Larman, Applying UML and Patterns (3rd ed), Prentice Hall, 2004 • NE Fenton & SL Pfleeger, Software Metrics: A Rigorous and Practical Approach (latest ed.), PWS Publishing, 1999 • B Hughes & M Cotterell, Software Project Management (2nd ed), McGraw-Hill, 1999 |
| <p>(ii) Core Texts</p> | | |
| <p>Reading Lists</p> | <p>Attached/Not Attached</p> | |

| | |
|--|---------------|
| Specification completed by: | Hai Wang |
| Date | 1st July 2010 |
| Date module approved by Learning and Teaching Committee(s) | |
| Date module approved by School Board(s) | |