**Liverpool** John Moores University

Warning: An incomplete or missing proforma may have resulted from system verification processing

| | |
|---|---|
| Title: | SECURE SOFTWARE DEVELOPMENT |
| Status: | Definitive |
| Code: | **5064COMP** (117506) |
| Version Start Date: | 01-08-2016 |
| Owning School/Faculty: | Computer Science |
| Teaching School/Faculty: | Computer Science |

| Team | Leader |
|---|---|
| Brett Lempereur | Y |
| Martin Randles | |

| | | | | | |
|---|---|---|---|---|---|
| **Academic Level:** | FHEQ5 | **Credit Value:** | 24 | **Total Delivered Hours:** | 74 |
| **Total Learning Hours:** | 240 | **Private Study:** | 166 | | |

**Delivery Options**

Course typically offered: Standard Year Long

| Component | Contact Hours |
|---|---|
| Lecture | 24 |
| Practical | 24 |
| Tutorial | 24 |

**Grading Basis:** 40 %

**Assessment Details**

| Category | Short Description | Description | Weighting (%) | Exam Duration |
|---|---|---|---|---|
| Portfolio | AS1 | Group software development and security task | 50 | |
| Exam | AS2 | Examination | 50 | 2 |

**Aims**

*To consolidate students' knowledge of the project management techniques used in software engineering projects.*

*To introduce the student to the models, tools and techniques of the development process for software systems, including formal principles of modelling, enabling students to apply these techniques in the analysis and design of systems.*
*To familiarise students with common software security problems and vulnerabilities, and the methods, tools and techniques that can be used during software development to prevent them, including formal techniques.*
*To provide students with an understanding of techniques that should be applied throughout the software development lifecycle in order to improve software security.*

## Learning Outcomes

After completing the module the student should be able to:

1 Evaluate the principles involved in project planning, scheduling and management, and apply these principles to practical examples.
2 Evaluate alternative models used for software engineering projects.
3 Analyse software security vulnerabilities and apply appropriate practical techniques to prevent them
4 Apply wide-ranging technical and conceptual security skills to the software development lifecycle.

## Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

| | | |
|---|---|---|
| Group task | 3 | 4 |
| Examination | 1 | 2 |

## Outline Syllabus

*-Characteristics of large-scale software systems projects. Team membership & activities.*
*-Software process models and Agile software development methodologies. CASE tool support.*
*-Software project planning - scheduling tools, milestones, option and risk analysis.*
*-Project monitoring and reviews, quality assurance.*
*-Validation and verification techniques. Formal specification models.*
*-Role of modelling in computing.  Engineering principles.*
*-Common software vulnerabilities (input validation errors, buffer overruns, SQL injection, API abuse etc.).*
*-Programming languages and security characteristics, decompiling and obfuscation.*
*-The secure software development lifecycle.*
*-Threat modelling.*
*-Formal techniques for vulnerability analysis (security requirements, code review, auditing, false positives vs. false negatives, etc.).*
*-Testing (stress testing, abuse cases, penetration testing), including practical experience of unit testing and fuzz testing.*
*-Networking vulnerabilities.*

*-Random number generation and cryptography.*
*-Secure deployment (software certification, patch management, software updates).*
*-General rules and guidelines; secure coding policies.*
*-Examples from computing are used throughout and practical exercises used to illustrate the applications of these concepts.*

## Learning Activities

Students will undertake a practical software development task contrasting three different methodologies: problem-decomposition, test-driven-development and the secure software development lifecycle.

Students will participate in lectures, tutorials, and lab sessions.

## Notes

This module introduces the issues involved in the management of software engineering projects, and their extension to incorporate secure software development techniques and the secure software lifecycle.