1. **Title of the module**

COMP5520 (CO552) Agile Development and Software Security A

1. **School or partner institution which will be responsible for management of the module**

Computing

1. **The level of the module (Level 4, Level 5, Level 6 or Level 7)**

Level 5

1. **The number of credits and the ECTS value which the module represents**

15 (7.5 ECTS credits)

1. **Which term(s) the module is to be taught in (or other teaching pattern)**

Autumn term

1. **Prerequisite and co-requisite modules**

COMP3200 (CO320) Introduction to Object Oriented Programming

COMP3230 (CO323) Databases and the Web

1. **The programmes of study to which the module contributes**

BSc Computing

BSc Computing (consultancy)

BSc Computer Science for Health

BSc Business Information Technology

Plus Year in Industry variants for all of the above

1. **The intended subject specific learning outcomes.
On successfully completing the module students will be able to:**

8.1 Understand and discuss the principles and practices employed in the production of a software system using an Agile incremental methodology. [A2, A4, B1, B2, B3, B5, C1, C4, C5, C7, D1]

8.2 Identify the differences in adopting an Agile approach to software development when compared to alternatives software development paradigms [B1, B3, B9, D2]

8.3 Understand the role and expected behaviour of key members of the development team and overall inherent values of the team [C5, D1]

8.4 Determine the product scope and requirements, and devise a corresponding product roadmap of backlogs, sprints and releases [A2, B3, B4, B5, C1, D7].

8.5 Describe the management of time and cost, quality and risk in Agile [C3, C5, C6],

8.6 Understand the headline issues of software security for users as well as system developers, consider how to ensure online and computer security in the work and home environments. Appreciate the risks inherent in software security breaches for users [B6, B10, C6].

8.7 To understand the role of abstraction in systems analysis [C1, C2].

8.8 To be able to specify information systems. This involves:

* Understanding project management [C4, C9]
* Carrying out a systems investigation [B6, C3, C4, C9, C11]
* UML modelling of requirements (Class, Use Case, Activity, Sequence and State diagrams) [B5, C4, C11]
1. **The intended generic learning outcomes.
On successfully completing the module students will be able to:**

9.1 Make effective use of IT facilities for scholarship and research. [D3]

9.2 Be able to manage their own time, learning and development. [D5]

9.3 Present and discuss a topic of study [B2, D2]

9.4 Recognise and be guided by professional, social and economic issues and guidelines. [B6, D1, D6]

1. **A synopsis of the curriculum**

The module studies in detail the activities and artefacts associated with the software development process as performed by a development team, particularly one that adopts an Agile methodology. Topics include

Concepts, principles, practice and philosophy of an Agile approach to software development

Collaboration: environment, programmer collaboration, team values, customer involvement, standards, reporting and professional responsibility

Planning: release and iteration/sprint planning, risk assessment, stories and estimating

Development: incremental requirements, customer tests, test-driven development, refactoring, simple design, incremental design and architecture, spike solutions, performance optimisation

Agile project management: roles, values and team philosophy; management of scope, time, cost, quality and risk.

Broad concepts of software security for users as well as system developers, including secure online behaviour and computer security both in work and home environments. Think about the risk of software security breaches for users.

Systems Analysis.

Object-Oriented methods and the use of UML for modelling of requirements (Class, Use Case, Activity, Sequence and State diagrams).

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**

Beck, K; Extreme Programming Explained: Embrace Change, 2e Addison-Wesley, 2005

Beck, K; Fowler, M; Planning Extreme Programming (XP),1e, Addison Wesley, 2001

Schwaber, K; Agile Project Management with Scrum, Microsoft Press, 2004

Layton, MC; Agile Project Management For Dummies, John Wiley & Sons, 2012

Pham, A; Pham, P-V Scrum in Action, 1e Delmar Cengage Learning, 2012

Cohn, M; Succeeding with Agile: Software Development Using Scrum, 1e, Addison Wesley, 2010

Fowler, M; UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edition, Addison-Wesley, 2004

Rumbaugh, J; Jacobson, I; Booch, G; The Unified Modeling Language Reference Manual, 2nd edition, Addison-Wesley, 2004

1. **Learning and teaching methods**

33 contact hours

117 private study hours

Total hours 150

1. **Assessment methods**13.1 Main assessment methods

Case Study 40% (25 hours)

Version control exercise 10% (5 hours)
2 hour Examination 50%

* 1. Reassessment methods: Like-for-like
1. **Map of module learning outcomes (sections 8 & 9) to learning and teaching methods (section12) and methods of assessment (section 13)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module learning outcome** | *8.1* | *8.2* | *8.3* | *8.4* | *8.5* | *8.6* | *8.7* | *8.8* | *9.1* | *9.2* | *9.3* | *9.4* |
| **Learning/ teaching method** |  |  |  |  |  |  |  |  |  |  |  |  |
| Private Study | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |
| *Lectures* | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |  |  |  | **x** |
| *Seminars* | **x** |  | **x** | **x** |  | **x** | **x** | **x** | **x** |  | **x** | **x** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |  |  |
| *Case study* | **x** | **x** | **x** | **x** |  |  | **x** | **x** | **x** | **x** | **x** | **x** |
| *Version control exercise* | **x** |  |  |  | **x** |  |  |  | **x** |  |  |  |
| *Examination* | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |  | **x** | **x** | **x** |

1. **Inclusive module design**

The School recognises and has embedded the expectations of current equality legislation, by ensuring that the module is as accessible as possible by design. Additional alternative arrangements for students with Inclusive Learning Plans (ILPs)/declared disabilities will be made on an individual basis, in consultation with the relevant policies and support services.

The inclusive practices in the guidance (see Annex B Appendix A) have been considered in order to support all students in the following areas:

a) Accessible resources and curriculum

b) Learning, teaching and assessment methods

1. **Campus(es) or centre(s) where module will be delivered**

Medway

1. **Internationalisation**

The topics addressed by this module relate to a field which is of international importance, given the global role of software engineering in today's technological innovation. The modelling techniques and software development methodologies covered by this module are international, being identical worldwide and independent of traditional spoken language.

**FACULTIES SUPPORT OFFICE USE ONLY**

**Revision record – all revisions must be recorded in the grid and full details of the change retained in the appropriate committee records.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date approved | Major/minor revision | Start date of the delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
|  |  |  |  |  |
|  |  |  |  |  |