SOFTWARE ENGINEERING
UNIT SYLLABUS

DEPARTMENT	Computer Science	**PROGRAMME**	BSc (Hons) in Computer Science
UNIT CODE	CCP3500	**UNIT TITLE**	Software Engineering
CREDITS	10	**SEMESTER/SESSION**	Autumn 2019
LEVEL OF STUDY	3rd	**STAFF OFFICE**	Athens
STAFF	Dr. Dranidis Dr. Gaviotis	**SKYPE NAME**	dimitris.dranidis ioanis.gaviotis
E-MAIL	dranidis@citycollege.sheffield.eu igaviotis@athtech.gr		

UNIT DESCRIPTION

The unit aims to develop an understanding of the problems involved in the development of high-quality software products and appreciation of the methodologies, techniques, and tools necessary to develop such systems efficiently. The emphasis of the course is placed on project management, metrics and project estimation, risk management, quality issues, testing, and contemporary software engineering topics, such as extreme programming and refactoring.

AIMS

This unit aims to:

A1 Present and discuss traditional software development process methodologies
A2 Introduce software estimation techniques and metrics as well as project management techniques
A3 Present risk management techniques and discuss the importance of software quality assurance
A4 Present testing techniques
A5 Introduce and discuss agile methodologies, such as extreme programming

LEARNING OUTCOMES

By the end of the unit, a student will be able to:		Link to aims
LO1	Given a problem, illustrate an appropriate software development process	A1, A5
LO2	Evaluate a software development process	A1, A5
LO3	Estimate the size, effort and time of a project	A2
LO4	Illustrate the role of metrics in software development	A2
LO5	Create a project plan by applying project management techniques	A2
LO6	Outline the risks involved in a project and construct a risk management and mitigation plan	A3
LO7	Employ techniques for quality assurance, evaluate systems in terms of quality	A3
LO8	Apply testing techniques and construct test cases based on white box and black testing techniques	A4

TEACHING & LEARNING METHODS Total Hours: 33

The following teaching & learning methods will be employed:

- There will be Weekly lectures and block teaching sessions. Lectures will include discussions of relevant case studies, participating in educational debates, answering quizzes and solving exercises.
- The practical assignments of the unit aim to help the students to develop skills in project management activities such as planning and estimation. The acquired skills are considered very useful for their work in their individual project.

ASSESSMENT METHODS				
Type #	Students will be assessed by:	Submission Week	% contribution	LOs assessed
C1	Software Project Estimation and Management Techniques	W8	40	LO1-5
E	Formal Examination	-	60	LO1-8

FEEDBACK PROVISION

The following methods will be used to provide formative and summative feedback to students:

- Formative verbal feedback: during advising sessions (personal support through open-door policy)
- Formative verbal feedback: during classroom discussions and quizzes
- Summative written feedback: following the submission of the project assignment (C1)
- Formative and summative feedback: during and after the Problems & Programmers game (C2)
- Summative written feedback following the formal examination (E)

The Feedback Handbook found at https://goo.gl/Zy2roA aims to give you a better understanding of feedback; what it's for and how to use it.

RECOMMENDED TEXTBOOK(S)

- Roger Pressman, *Software Engineering*, A Practitioner's Approach, McGraw-Hill, 8th edition, 2018

LIST OF REFERENCES / ADDITIONAL RECOMMENDED READING

- Ian Sommerville, *Software Engineering*, Addison-Wesley, 9th edition, 2011
- Shari Lawrence Pfleeger, *Software Engineering: Theory and Practice,* Prentice Hall, 2nd edition, 2001
- Kent Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999
- Martin Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999
- Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, anniversary edition (2nd edition), Addison-Wesley, 1995
- Steve McConnell, *Code Complete,* Microsoft Press, 2nd edition, 2004

OUTLINE

Week #1	**Introduction to Software Engineering** Software characteristics, software crisis, software engineering
Week #2	**Software Engineering process** Prescriptive models: Waterfall, incremental, evolutionary
Week #3	**Project management: software metrics** Measures, metrics, indicators, software measurement, measuring quality
Week #4	**Project estimation** Software project estimation, Function Points, algorithmic estimation models: COCOMO
Week #5	**Project planning** Roles, task analysis, time scheduling, monitoring resources with tools
Week #6	**Risk management** Software risk analysis, estimation & monitoring
Week #7	**Software quality assurance** Quality concepts, quality control, quality standards: ISO, CMM
Week #8	**Software testing I** Testing strategies, unit testing, integration testing, regression testing, system testing
Week #9	**Software testing II** Verification & validation, test case design, white box testing, black box testing
Week #10	**Agile process models** Light-weight methodologies, eXtreme Programming practices
Week #11	**Revision**

SHEFFIELD GRADUATE BADGES

This unit contributes to the following Sheffield Graduate attributes as described in https://www.sheffield.ac.uk/sheffieldgraduate/studentattributes :