

Module Title	Software Engineering
Level	5
Reference Code	CSI_5_SFE
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-Requisites	None
Co-requisites	None
Excluded combinations	None
Module coordinator	TBC
Division	Division of Computer Science and Informatics
Short Description	This module covers all aspects of Software Engineering, the application of systematic, disciplined approaches to the development, testing, maintenance and ongoing development of software. It covers the fundamental requirements and established concepts necessary for effective software development projects and investigates diverse methodologies we seek to attain them.
Aims	This module aims to give students a thorough understanding of the methodologies and techniques that software engineers use to ensure the successful design, delivery and maintenance of complex yet reliable software systems. Students will gain experience of the tools and notations used and an overview of the entire process of system design from inception and requirements elicitation to delivery, testing and ongoing maintenance.
Learning Outcomes	LO1: Knowledge and Understanding - Apply the main practices, techniques, notations and methodologies used in software engineering. LO2: Intellectual Skills - Critically evaluate and compare software engineering approaches and techniques. (Maps to: BCS 2.2.1 a1-a5, a7-a9;) LO3: Practical Skills - Identify requirements and produce corresponding specifications and development plans for complex software systems (Maps to: BCS 2.2.1; b1-b4;) LO4: Transferable Skills - Document systems in both technical and non-technical terms. Work effectively in teams. (Course CS d1, d2, d3)
Employability	The specification and development of computing and information systems is now required in every field of human activity. The ability to recognise and identify the requirements of systems and match them to corresponding specifications is of value to any employer who needs to select, commission and use systems. In developer roles, a good grasp of software engineering principles is essential to plan, manage and successfully complete projects, whilst these skills also have more general organisational and strategic applications in general project management scenarios.
Teaching and Learning Pattern	The module will be delivered using a combination of lecture/seminar sessions and computer lab/workshop sessions. The lecture/seminars will consist of the delivery, discussion and intellectual investigation of factual and conceptual material. The laboratory sessions will consist of practical exercises using relevant technologies and provide opportunities for students to develop their understanding through independent experimentation.

Indicative Content	• Introduction to software process models (e.g., waterfall, incremental, agile) • Software requirements elicitation • Describing functional requirements using, for example, use cases or user stories • Describing system data using, for example, class diagrams or entity-relationship diagrams • Relationships between requirements and designs: transformation of models, design of contracts, invariants • Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented • Software configuration management and version control • Software reliability engineering concepts • Verification and validation concepts • Software change, concerns and concern location, refactoring
Assessment Elements & weightings	COURSEWORK 100% Summative Assessment This coursework is expected to consist of a group-based coursework. Students should develop their ability to work with their colleagues towards a common goal. This should involve developing a software system and produce software process artefacts like a requirements specification, design and architecture documents, and a test plan. (LO1-LO4) (Maps to: BCS 2.2.1 a1-a9; b1-b4; c1-c2; 2.2.3 a4-a6) Formative Assessment Skills for the summative assessment will be embedded throughout formative opportunities in Lectures and Workshops. Formative assessment will take different forms, such as: • interactive revision quizzes • think-pair-share concept and class discussions • verbal feedback on tutorial activities • observation and questioning to provide instant feedback as the student takes part in learning activities • self and peer assessment
Indicative Sources (Reading lists)	Core: • Sommerville, I. (2015) <i>Software Engineering</i>. Pearson; ISBN 1292096136 Optional: • Mistrik, I., Bahsoon, R., Ali, N., Heisel, M., & Maxim, B. (2017). Software architecture for big data and the cloud. ProQuest Ebook Central https://ebookcentral.proquest.com • Stephens, R. (2015) <i>Beginning Software Engineering</i>. Wrox; ISBN 1118969146 • Pressman, R.S. (2014) <i>Software Engineering: A Practitioner's Approach</i>. McGraw-Hill Higher Education; ISBN 1259253155