

# CS261-15 Software Engineering

**26/27**

**Department**

Computer Science

**Level**

Undergraduate Level 2

**Module leader**

Ian Saunders

**Credit value**

15

**Module duration**

10 weeks

**Assessment**

Multiple

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

The aim of the module is to provide students with a theoretical understanding and practical experience of current best practice in software engineering.

This module is only available to students in the second year of their degree and is not available as an unusual option to students in other years of study.

### Module aims

The aim of the module is to provide students with a theoretical understanding and practical experience of current best practice in software engineering. The module concentrates on the application of software engineering principles to the development of a significant software system, with an emphasis on design quality, technical evaluation, team working and project management.

### Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

Topic areas addressed are software development processes, feasibility and requirements analysis, modelling and design, implementation and design patterns, testing and analysis,

maintenance and evolution. Each area will be covered from a theoretical and practical perspective, with a particular emphasis on concepts such as:

- Object oriented modelling and design: objects, classes, inheritance, polymorphism, aggregation, abstract classes.
- Formal notations for expressing issues relating to the analysis, design and implementation of systems, including classes, objects, sequences, packages, collaborations, activities, system states and components.
- Classical software engineering principles, including software processes, waterfall model, software life cycles, cost-estimation, safety critical systems, software testing and maintenance.
- Human computer interaction: human information processing models and memory, learning and skill acquisition; interaction styles; the graphical user interface; task analysis and user centred design; evaluation techniques.
- Techniques relating to the elicitation of requirements, including issues in process engineering, user-centred design and customer management.
- Architectural design and implementation technology selection, including the application of design patterns and frameworks for system design.
- Levels of system and performance testing, including unit, integration, system and user / acceptance testing.
- Software process issues relating system documentation, evolution and maintenance.

## Learning outcomes

By the end of the module, students should be able to:

- - Understand the importance of the stages in the software life cycle, including a range of software development development methodologies.
- - Design object oriented software using with the aid of a formal system modelling notation.
- - Understand the principles of graphical user interface design.
- - Understand the principles of user-centred design.
- - Critically evaluate their experience of practical software development with regard to project management, software processes and technical accomplishment.
- - Understand, through experience, the practical challenges associated with the development of a significant software system, including a graphical user interface.
- - Understand, through experience, the practical challenges associated with working as a member of a software development team.

## Indicative reading list

[Reading lists can be found in Talis](#)

[Specific reading list for the module](#)

## Subject specific skills

Understand the importance of the stages in the software life cycle, including a range of software

development methodologies.

Design object oriented software using with the aid of a formal system modelling notation

Understand the principles of graphical user interface design

Understand the principles of user-centred design

Critically evaluate their experience of practical software development with regard to project management, software processes and technical accomplishment

Understand, through experience, the practical challenges associated with the development of a significant software system, including a graphical user interface

Understand, through experience, the practical challenges associated with working as a member of a software development team.

## Transferable skills

Communication skills (written and verbal)

Presentation skills

Team work

Team based coding

Problem analysis

Critical evaluation skills

---

## Study

### Study time

Type	Required
Lectures	15 sessions of 1 hour (10%)
Seminars	3 sessions of 1 hour (2%)
Project supervision	4 sessions of 30 minutes (1%)
Private study	77 hours 30 minutes (51%)
Assessment	52 hours 30 minutes (35%)
Total	150 hours

### Private study description

Independent study includes both revision of the material and the group project. The project has a number of factors, including a variety of reports and software tasks. Engaging with these tasks, and revising the expected approaches and techniques of these tasks, help to reinforce the material from the lectures. The remaining time should be spent on revision and performing the background reading.

### Costs

No further costs have been identified for this module.

---

## Assessment

You do not need to pass all assessment components to pass the module.

Students can register for this module without taking any assessment.

### Assessment group A

	<b>Weighting</b>	<b>Study time</b>	<b>Eligible for self-certification</b>
Group software development project - Requirements and Design Group software development project - Requirements and Design.	20%	30 hours	No
Group Video Demo Presentation Video presentation, between 5 and 10 minutes, for the group to showcase the demo of their software project.	15%	22 hours 30 minutes	No
Peer Assessment Report Student report submission evaluating other group members. Mark is derived for each group member from other member's submitted reports.	15%		No
Final Individual Report Final individual report where students can evaluate their group project, demonstrating practical application of the methods and processes discussed within the lecture content.	50%		No

### Assessment group R2

	<b>Weighting</b>	<b>Study time</b>	<b>Eligible for self-certification</b>
Final Individual Report Resit report on project.	100%		No

### Feedback on assessment

Tabula

---

## Availability

## Courses

This module is Core for:

- Year 2 of UCSA-G500 Undergraduate Computer Science
- UCSA-G503 Undergraduate Computer Science MEng
  - Year 2 of G500 Computer Science
  - Year 2 of G503 Computer Science MEng
- Year 2 of UCSA-I1N1 Undergraduate Computer Science with Business Studies
- Year 2 of UCSA-G406 Undergraduate Computer Systems Engineering
- Year 2 of UCSA-G408 Undergraduate Computer Systems Engineering
- Year 2 of USTA-G302 Undergraduate Data Science
- Year 2 of USTA-G305 Undergraduate Data Science (MSci) (with Intercalated Year)

This module is Option list B for:

- Year 2 of UCSA-G4G1 Undergraduate Discrete Mathematics
- UCSA-G4G3 Undergraduate Discrete Mathematics
  - Year 2 of G4G1 Discrete Mathematics
  - Year 2 of G4G3 Discrete Mathematics