

2024 / 25

School of Science and Computing

☎ +353 (0)51 302037

✉ [Eleanor.Reade@setu.ie](mailto:Eleanor.Reade@setu.ie)

🌐 [www.wit.ie/schools/science\\_computing](http://www.wit.ie/schools/science_computing)



**SE  
TU**

Ollscoil  
Teicneolaíochta  
an Oirdheiscirt

South East  
Technological  
University

## Module Descriptor

---

# Software Engineering Practice (Computing and Mathematics)

# Software Engineering Practice (A15081)

**Short Title:** Software Engineering Practice  
**Department:** Computing and Mathematics  
**Credits:** 5

**Level:** Introductory

## Description of Module / Aims

The objective of this module is to provide students with the understanding of techniques and methods used to develop reliable quality software. Students will also research, experiment and utilize various tools in order to explore the typical activities of modern software engineering practices. These tools will also be used to deliver a small software application.

## Programmes

			stage/semester/status
COMP-0640	BSc (Hons) in Applied Computing (International) (WD_KACCM_BI)		3 / 6 / M
COMP-0640	BSc (Hons) in Applied Computing (WD_KACCM_B)		2 / 4 / E
COMP-0640	BSc (Hons) in Applied Computing (WD_KCOMP_B)		2 / 4 / M
COMP-0640	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)		2 / 4 / M
COMP-0640	BSc (Hons) in Computer Science (WD_KCMSC_B)		2 / 4 / E

## Indicative Content

- Project Management
- Software Processes
- Requirements Elicitation & Modelling
- Automated Project Build and Testing
- Continuous Integration & Deployment
- Project Scaffolding Tools

## Learning Outcomes

*On successful completion of this module, a student will be able to:*

1. Compare different software processes and select a suitable process for the design of a particular software system.
2. Explain the importance of project planning and create a project workflow using appropriate tools.
3. Describe software requirements and illustrate the processes involved in discovering these requirements.
4. Demonstrate an understanding of the different models and activities in the object oriented design process.
5. Design and automate an appropriate test driven strategy for a software development project.
6. Operate a Continuous Integration (CI) practice in collaborative software projects.
7. Illustrate the advantages of Continuous Delivery (CD).

## Learning and Teaching Methods

- This module will be presented using a combination of lectures and practical classes.
- The lectures will be used to introduce new topics and their related concepts.
- The lab-based practicals, building on the theoretical knowledge from lectures, provide exposure to the automation tools and practical skills required to support a project life-cycle.
- The practical content will use industry standard technologies, tools and techniques.
- Students will be encouraged to enhance their lab work and assessment submissions using self-directed research into state-of-the-art of software engineering practices.

## Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	12	
Practical	36	
Independent Learning	87	

## Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Group Project	60%	2,3,4,5,6
In-Class Assessment	40%	1,2,3,7

## Assessment Criteria

<40%: Unable to interpret and describe key concepts of software engineering practices.

40%–49%: Be able to interpret and describe key concepts of software engineering practices.

50%–59%: Ability to discuss key concepts of software engineering and the ability to discover and integrate related knowledge in other knowledge domains.

60%–69%: Be able to solve problems within software engineering by experimenting with the appropriate skills and tools.

70%–100%: All the above to an excellent level. Be able to analyse and design solutions to a high standard for a range of both complex and unforeseen problems through the use and modification of appropriate skills and tools.

## Supplementary Material(s)

- "Continuous Delivery." <https://www.thoughtworks.com/continuous-delivery>. <https://www.thoughtworks.com/continuous-delivery>
- Pressman, R. and B.R. Maxim. *Software Engineering: A Practitioner's Approach*. 8th. New York: McGraw-Hill Higher Education, 2014.
- Sommerville, I. *Software Engineering*. 10th. Boston: Pearson, 2015.

## Requested Resources

- Room Type: Computer Lab