

Module: Software Testing 261 (Deaf)

Module name:	Software Testing 261 (Deaf)
Code:	D-SWT261
NQF level:	6
Type:	Fundamental – Diploma in Information Technology (Software Development Stream)
Contact time:	24 hours
Structured time:	4 hours
Self-directed time:	12 hours
Notional hours:	40 hours
Credits:	4
Prerequisites:	Programming 261 (Deaf)

Purpose

The main focus of this course is on realistic, pragmatic steps for rigorous and organized software testing. It clarifies testing terminology and covers the different types of testing performed at each phase of the software lifecycle together with the issues involved in these types of testing. The course will discuss how tests can be derived from requirements and specifications, design artefacts, or the source code, and introduce proper testing tools. At the end of the course, students will have an appreciation of a range of testing techniques, and an understanding of rigorous testing theory. They will be able to select an appropriate testing strategy, devise suitable test cases, and formulate correctness hypotheses.

Outcomes

Upon successful completion of this module, the student will be able to demonstrate:

- Detailed knowledge of the main areas of software testing, including an understanding of and the ability to apply the key terms, concepts, standards, principles and theories of software testing to new but relevant contexts
- Knowledge of specific testing levels and testing techniques and how that knowledge relates to other stages of the software development lifecycle.
- The ability to evaluate, select and apply appropriate testing models, processes and practices, and strategies in the context of the software development lifecycle, for both complete lifecycles and individual phases of the lifecycle
- The ability to evaluate different sources of information, to select information appropriate to the task and to design specific and measurable test cases to ensure coverage and traceability to requirements
- The ability to present and communicate complex information regarding testing results and project status reliably and coherently using appropriate professional problem reporting techniques, metrics, and testing status reports to colleagues, managers, and end users by producing appropriate documentation for managing each stage of the testing process

Assessment

Assessment is performed using a variety of instruments:

- Continuous evaluation of theoretical work through a written assignment and a summative test.
- Final assessment through an examination.
- The assignments or projects collectively will count 20% of your class mark.
- All tests will collectively account for 80% of your class mark.
- Your class mark contributes 30% towards your final mark for the subject, while the final assessment accounts for 70% of your final mark.

Teaching and Learning

Learning materials

Prescribed books (EBSCO)

- 📖 **Rex Black et al. (2017) Agile Testing Foundations : An ISTQB Foundation Level Agile Tester Guide. Swindon, UK: BCS, The Chartered Institute for IT.**
- 📖 **Hass, A. M. J. (2014) Guide to Advanced Software Testing. Boston: Artech House.**
- 📖 **Rex Black (2011) Pragmatic Software Testing : Becoming an Effective and Efficient Test Professional. Chichester: Wiley.**
- 📖 **Kshirasagar Naik and Priyadarshi Tripathy (2008) Software Testing and Quality Assurance : Theory and Practice. Hoboken, N.J.: Wiley-Spektrum.**

Learning activities

Learning will be facilitated by the lecturer with student centred activities that involve problem-based learning where pupils are presented with challenges that replicate the situation in the real-world environment. This will be achieved through a combination between presentation of theoretical concepts, guided exercises, group work and discussions together with one mandatory assignment to be completed during the module.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		14.0		6.0
Formative feedback		2.0		
Project				
Assignment	1			3.0
Test	1		2.0	5.0
Exam	1		2.0	6.0
		16.0	4.0	20.0

Syllabus

- Fundamentals of testing and the test process
- Testing in the Software Life Cycle: the general V-Model, and test levels, including: unit testing; integration testing; system testing; acceptance testing; performance, stress, and configuration testing;
- Test management, including: test planning, organisation and strategy, test progress monitoring and control, test reporting