

Unit summary

This unit is about the fundamentals of software engineering and providing a framework which allows a disciplined approach to the development and maintenance of a range of high quality software.

Aims

The unit aims to equip the candidate with the knowledge and skills to monitor and measure aspects of the software process, and to implement mechanisms leading to the development of high quality software.

Prerequisites

It is expected that candidates will have a working knowledge of the materials in the four compulsory papers of the Certificate examination and subject 9107-108 Software and information systems engineering.

Learning outcomes

There are **two** outcomes to this unit. The candidate will be able to:

- Understand the nature of the software process and make selection of different approaches to its realization in particular situations
- Validate, verify and manage software

Guided learning hours

It is recommended that 300 hours should be allocated for this unit. 120 of those hours are actual taught hours. This may be on a full time or part time basis.

Key Skills

This unit contributes towards the Key Skills in the following areas:

IT4.1

Develop a strategy for using IT skills over an extended period of time.

IT4.2

Monitor progress and adapt your strategy, as necessary, to achieve the quality of outcomes required in work involving the use of IT for **two** different, complex purposes.

IT4.3

Evaluate your overall strategy and present the outcomes from your work using at least **one** presentation, showing integration of text, images and number.

PS4.1

Develop a strategy for using skills in problem solving over an extended period of time.

PS4.2

Monitor progress and adapt your strategy, as necessary, to achieve the quality of outcomes required when tackling **one** complex problem with at least three options.

PS4.3

Evaluate your overall strategy and present the outcomes from your work using a variety of methods.

Occupational Standards

This unit has been mapped to the following National Occupational Standards:

- 1.1.1 Identify the requirements of clients for engineering products or processes
- 1.1.2 Produce specifications for engineering products or processes
- 1.3.1 Undertake research into engineering products or processes
- 1.3.2 Evaluate the results of research
- 1.4.1 Establish a design brief for engineering products or processes
- 2.1.1 Determine the production requirements of engineering products and processes
- 3.1.1 Determine the installation requirements for engineering products or processes
- 3.1.2 Specify installation methods and procedures to achieve installation requirements
- 3.4.2 Configure engineering products or processes
- 4.1.1 Determine the operational requirements of engineering products or processes
- 4.3.1 Monitor operational processes
- 5.1.1 Determine the maintenance requirements of engineering products or procedures
- 5.1.2 Specify maintenance methods and procedures to achieve maintenance requirements
- 5.2.1 Implement maintenance methods and procedures
- 6.2.1 Assure the quality of engineering products or processes
- 7.1.1 Develop objectives for projects
- 7.2.3 Evaluate projects
- 8.1.1 Maintain and develop own engineering expertise

Unit 229

Software engineering

Outcome 1

Understand the nature of the software process and make selection of different approaches to its realization in particular situations

Knowledge requirements

The candidate knows how to:

- 1 Software Lifecycle
- 2 appraise the standard software life cycle model
 - a requirements
 - b specification
 - c architectural design
 - d detail design
 - e implementation and testing
 - f coding
 - g maintenance
- 3 appraise alternative models
 - a spiral model
 - b prototyping model
 - c transformational model
- 4 evaluate the strengths and weaknesses of the above approaches to life cycle modelling, particularly with reference to the object-oriented design paradigm
- 5 Software Process
- 6 understand the concept of the software process
- 7 appraise the SEI five level maturity model
- 8 Software Specification
- 9 understand the functional specification of software
- 10 understand the non-functional specification of software
- 11 understand requirements analysis and relationship with systems engineering and systems analysis and design
- 12 appraise systems modelling
- 13 discriminate between formal and informal approaches to describing specifications and their relative merits
- 14 assess different approaches to formal specifications
 - a benefits
 - b drawbacks
- 15 understand a common formal specification language, for example
 - a Z or
 - b VDM

- 16 produce a specification of a simple system using a formal specification language
- 17 Software Design
- 18 analyse software design
- 19 characteristics of good design
- 20 different approaches to architectural design
- 21 undertake a detailed study of each of the following approaches to software design
 - a object oriented
 - b data flow methods
 - c real-time methods
- 22 CASE tools to support these activities
- 23 Interface Issues
- 24 understand the principles of interface design including the relative merits of
 - a graphical user interface systems
 - b command interfaces
- 25 undertake user interface evaluation with respect to these principles
- 26 understand tool support for interface design
- 27 Implementation Issues
 - a choice of programming language
 - b choice of programming project support environment
 - c selection of appropriate tools and factors influencing their choice
 - d understand
 - i strong typing
 - ii safe programming constructs
 - iii exceptions
 - iv fault tolerance
 - v information hiding to combat complexity
 - vi separate compilation
 - vii concurrency

Knowledge requirements

The candidate knows how to:

- 1 Validation and Verification
 - a validate and verify software
 - i terminology
 - ii use of reviews and walkthroughs
 - iii steps to be taken to ensure the effectiveness of approaches
 - b use validation and verification throughout the life cycle
 - c understand different approaches to testing, including
 - i black box or functional testing
 - ii white box or structural testing
 - iii path coverage
 - iv dynamic analysers
- 2 undertake program verification using verification conditions and loop invariants in providing proofs of the correctness of simple programs that involve
 - a assignment statements
 - b conditional statements
 - c simple while loops
- 3 evaluate the strengths and weaknesses of different approaches to verification and validation
- 4 Software Management
 - a manage software
 - i aims and purpose
 - ii planning and scheduling
 - iii risk identification and analysis
 - b understand software re-engineering and re-use
 - c cope with change in all its forms
 - i personnel
 - ii improvements and defect removal
 - iii requirements
 - iv specifications
 - v hardware
 - vi tools
 - vii environment
 - d assess team selection methods and team building techniques

- e appreciate software engineering standards
 - i coding
 - ii company specific
 - iii national and international
- 5 understand the role and purpose of standards
- 6 understand software metrics and related tools to assist management
- 7 assess software cost estimation models including COCOMO
- 8 understand software quality assurance
- 9 assess the role of management in the review process
- 10 Configuration and Version Control
- 11 explain configuration management and version control
- 12 state the main activities involved in configuration management and version control
- 13 use tools to support software engineering activities including
 - a make
 - b RCS
 - c SCCS
- 14 have a disciplined approach to the above activities based on the use of appropriate tools
 - a change control boards
 - b impact analysis
 - c appropriate statistics gathering
 - d release of changes

Unit 229 Software engineering

Recommended reading list

Core texts	Author(s)	Publisher	ISBN
An Introduction to Discrete Mathematics, Formal System Specification and Z	Ince	Clarendon Press	0198538367
Managing the Software Process	Watts Humphrey	Addison-Wesley	0201180952
Software Engineering	Sommerville	Addison-Wesley	0321210263
Software Engineering with Java	Schach	Irwin	0256241678
Other useful texts			
Program Derivation	Geoff Dromey	Addison-Wesley	0201416247
Software Development with Z	Wordsworth	Addison-Wesley	0201627574
Software Engineering- A European Perspective	Richard, Thayer, McGetterick	IEEE Computer Press	0818691174
Software Engineering-Principles and Practice	Van Vilet	John Wiley	0471975087
Systematic Software Development using VDM	Jones	Prentice Hall	0138804362
Z - An Introduction to Formal Methods	Diller	John Wiley	0471939730
Program Construction and Verification	Backhouse	Prentice Hall	0137291469 o/p
Software Configuration Management	Babich	Addison-Wesley	0201101610 o/p