



---

Unit title:	Software applications testing
Unit number:	14
Level:	4
Credit value:	15
Guided learning hours:	60
Unit reference number:	L/601/1984

---

### **UNIT AIM AND PURPOSE**

To help learners understand and use the principles of software applications testing as part of the development of commercial quality software. The unit enables learners to apply structured approaches to testing and managing application development.

## **LEARNING OUTCOMES AND ASSESSMENT CRITERIA**

A pass grade is achieved by meeting **all** the requirements in the assessment criteria.

<b>Learning Outcome (LO)</b>	<b>Pass</b>
The Learner will:	The Learner can:
LO1 Understand the principles of software application testing	1.1 evaluate testing techniques applicable to the testing opportunity 1.2 compare the relative benefits of different testing methodologies 1.3 justify a proposed testing methodology
LO2 Be able to design test strategies	2.1 design a test strategy for a given testing opportunity 2.2 design a test plan for a given testing opportunity 2.3 justify the test plan proposition and testing strategy
LO3 Be able to implement test plans	3.1 implement a test plan based on a given testing opportunity
LO4 Be able to evaluate test plans	4.1 critically review the test outcomes 4.2 justify the validity of the test and identify any potential issues

## **GRADING CRITERIA**

A merit grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors.

A distinction grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors **and** the distinction descriptors.

<b>Merit Criteria (M1, M2, M3)</b>	<b>Distinction Criteria (D1, D2, D3)</b>
(M1, M2, and M3 are mandatory to achieve a merit grade. Each must be achieved at least once per unit to achieve a merit grade.)	(D1, D2, and D3 are mandatory to achieve a distinction grade. Each must be achieved at least once per unit to achieve a distinction grade.)  (In order to achieve a distinction grade, all merit criteria must also have been achieved.)
<b>MANDATORY TO ACHIEVE A MERIT GRADE</b>	<b>MANDATORY TO ACHIEVE A DISTINCTION GRADE</b>
M1 Analyse concepts, theories or principles to formulate own responses to situations.	D1 Evaluate approaches to develop strategies in response to actual or anticipated situations.
M2 Analyse own knowledge, understanding and skills to define areas for development.	D2 Evaluate and apply strategies to develop own knowledge, understanding and skills.
M3 Exercise autonomy and judgement when implementing established courses of action.	D3 Determine, direct and communicate new courses of action.

## **TEACHING CONTENT**

The Teaching Content describes what has to be taught to cover **all** Learning Outcomes.

Learners must be able to apply relevant examples to their work although these do not have to be the same as the examples specified.

### **LO1 Understand the principles of software application testing**

Learners will be taught the principles which underpin the design, running and management of testing activity during the creation of commercial quality software.

Testing strategy and specification	<ul style="list-style-type: none"><li>identifying and capturing user needs and defining requirements, detail of expected outcomes, expected timeline</li></ul>
Prototyping	<ul style="list-style-type: none"><li>using dummy data and checking expected outcomes</li></ul>
Requirements	<ul style="list-style-type: none"><li>access to resources (people, hardware, testing software)</li></ul>
Techniques	<ul style="list-style-type: none"><li>unit testing: black box, white (or glass) box, grey box, functional</li><li>system testing: regression, integration, whole system, interface</li><li>User Acceptance Testing (UAT) – interface, user story (Agile development tools), operation and contract testing</li></ul>
Methods and Approaches	<ul style="list-style-type: none"><li>Component-based, code walkthroughs and smoke tests, event only, alpha, beta, test driven development and Agile methods</li></ul>
Maintenance	<ul style="list-style-type: none"><li>change management process, reviews, time based, stress and predictive maintenance, use of performance data for monitoring maintenance</li></ul>
Documentation	<ul style="list-style-type: none"><li>specifications (business requirements and technical), user requirements capture and creation of associated test plans and logs</li></ul>

### **LO2 Be able to design test strategies**

Learners will be taught how to specify both scope and activities (e.g. timescales and project plan), provide a rationale for selecting a particular strategy, give specific details of functionality to be tested. Learners should be able to identify maintenance issues and include these as part of a testing strategy.

Designing test plans	<ul style="list-style-type: none"> <li>specifying examples of types of data (normal, erroneous, extreme)</li> <li>detailing expected outcomes such as expected, unexpected, priority and estimation of fix resources (time, cost, use of test resources), new information collected</li> </ul>
Techniques	<ul style="list-style-type: none"> <li>types of testing such as black box, white (or glass) box, creating test cases, use of test-driven development strategies, code walkthroughs</li> </ul>
Version control	<ul style="list-style-type: none"> <li>alpha testing in closed test, beta test involving end users, version cycles, bug fixing and use of version control standards and software</li> </ul>

### **LO3 Be able to implement test plans**

Change management	<ul style="list-style-type: none"> <li>processes to track and manage testing and bug fixing (manage reporting process, bug reporting, bug response and fix process, allocating resources)</li> </ul>
Tools	<ul style="list-style-type: none"> <li>testing tools such as Bugzilla</li> <li>functional testing tools such as JUnit</li> <li>integrated tools such as Jubula (Eclipse) or Microsoft Test Manager (Visual Studio)</li> </ul>
Tracking	<ul style="list-style-type: none"> <li>prioritisation of tests and fixes</li> <li>categorisation of issues</li> <li>response (times, format of response – inclusion in future release, deprecation of functionality etc.)</li> </ul>
Quality Assurance	Learners should be aware of the quality assurance process including adjusting timelines, time management, allocation of resources, feedback to customer, tracking adherence to standards, managing variances in performance

### **LO4 Be able to evaluate test plans**

Evaluation	Effectiveness and usability of test strategies including the importance of issues such as timeliness in identifying possible improvements in architecture and design, reflection on product and the role of testing in producing high quality software. For a given strategy, describe the usefulness to self, usefulness to others, usefulness for customers of the testing activity.
------------	--

## **GUIDANCE**

### **Delivery guidance**

It will be beneficial to deliver this unit in a way that uses actual events, industry forecasts or sector specific contexts which offer the learner the opportunity to explore, develop and apply the fundamental principles of the sector or subject area. Typical delivery contexts could include the design and delivery of a complex piece of software in their own work environment.

Learners will benefit from being encouraged to exercise autonomy and judgement to identify and provide a rationale for the test strategy they select. They will need to adapt their thinking and reach considered conclusions, when adjusting tests and making decisions on feasibility of making changes to their code based on test outcomes (based on a foundation of relevant knowledge, understanding and/or practical skills).

Learners would benefit from being presented with subject/sector-relevant problems from a variety of perspectives and from being given the opportunity to explore them using a variety of approaches and schools of thought. For example, the same project viewed from a traditional testing approach based on a Waterfall model of development and from an Agile perspective. This gives learners a chance to think about testing both as a phase of development or as part of every phase and every iteration of a project.

### **Assessment evidence guidance**

Evidence must be produced to show how a learner has met each of the Learning Outcomes. This evidence could take the form of assignments, project portfolios, presentations or, where appropriate, reflective accounts.

Where group work/activities contribute to assessment evidence, the individual contribution of each learner must be clearly identified.

All evidence must be available for the visiting moderator to review. Where learners are able to use real situations or observations from work placement, care should be taken to ensure that the record of observation accurately reflects the learner's performance. This should be signed, dated, and included in the evidence. It is best practice to record another individual's perspective of how a practical activity was carried out. Centres may wish to use a witness statement as a record of observation. This should be signed and dated and included in the evidence.

## **RESOURCES**

### **Books**

Kaner C., Falk J., Nguyen H., *Testing Computer Software* (2<sup>nd</sup> Ed), Thomson Press, 1999.

Whitaker J., *Exploratory Software Testing, Tips Tricks, Tours and Techniques to Guide Test Design*, Addison Wesley, 2009.

Hambling B. (Ed), Morgan P., Samaroo A., Thompson G., *Software Testing: An ISTQB-ISEB Foundation Guide*, British Computer Society, 2010.

### **Journals**

*Journal of Software Testing Reliability and Verification*

### **Websites**

[www.software-testing-training.com/](http://www.software-testing-training.com/)

[www.bugzilla.org/](http://www.bugzilla.org/)

<http://standards.ieee.org/> (standard 829 Standard for Software and System Test Documentation)

(if Shibboleth access not possible Wikipedia article is helpful

[http://en.wikipedia.org/wiki/IEEE\\_829](http://en.wikipedia.org/wiki/IEEE_829))