



Australian Government

Department of Education, Employment and Workplace Relations

ICAGAM511A Manage testing of games and interactive media

Release: 1

ICAGAM511A Manage testing of games and interactive media

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICAI1 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to manage the testing of games and interactive media to enable timely product release.

Application of the Unit

This unit applies to IT personnel who take responsibility for managing testing games and interactive media.

The management of the testing of a product directly impacts on the quality and time lines of delivery to market of a product. Good test management can deliver a quality product. Badly designed products take longer to test but can still be of quality, when partnered with good test management. Well-designed software and good test management go hand in hand to produce quality, timely software products.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

Elements and Performance Criteria

<p>1. Determine quality requirements statement which will enable product release</p>	<p>1.1 Review picture of product in market place to determine high-level requirements provided by product, for expected clients, considering all <i>types of requirements</i></p> <p>1.2 Define a releasable product in terms of <i>outstanding bugs</i>, to enable a <i>limited release</i> and a complete product release</p> <p>1.3 Summarise findings into a product release enabling quality requirements statement</p> <p>1.4 Confirm client and development, agreement on product release enabling quality requirements statement</p>
<p>2. Define test plan to enable testing of required quality as defined by agreed quality requirements statement</p>	<p>2.1 Determine expected <i>test cycles</i>, during the software development life cycle, considering what development methodology is in use and what the quality requirements are for product release</p> <p>2.2 Determine what <i>types of testing</i> will be performed during the test cycles to enable efficiency of processes and confirmation of quality requirements statement</p> <p>2.3 Determine what <i>testing methods</i> will be used to implement testing types defined for defined test cycles</p> <p>2.4 Determine <i>testing technique</i> to be used to determine test cases and analyse results</p> <p>2.5 Perform test cycle until a combination is found that provides an acceptable balance of cost, quality and risk, for upper management and development to agree to</p> <p>2.6 Selection <i>test-support software</i> to enable efficiency in testing and testing management</p> <p>2.7 Define <i>implementation details</i> for agreed testing and team responsible for testing management</p> <p>2.8 Define <i>reporting details</i> for testing throughout product life cycle to enable ongoing management of testing process</p> <p>2.9 Confirm test plan completeness using available <i>completeness techniques</i></p> <p>2.10 Confirm test plan with development and management</p>
<p>3. Install and configure test plan defined test support software</p>	<p>3.1 Install and configure bug tracking process and define bug description fields to maximise efficiency and minimise possibilities of bouncing bugs</p> <p>3.2 Install and configure test case management software</p> <p>3.3 Install and configure test cycle management and reporting software</p>

	3.4 Install and configure automated test tools
4. Manage testing process to enable defined quality requirements	<p>4.1 Manage and report on development of test cases</p> <p>4.2 Manage and report on test cycle status</p> <p>4.3 Manage and report on outstanding bug status</p> <p>4.4 Manage and report on status of product testing related to product release enabling quality requirements statement</p> <p>4.5 Update test plan and schedule to deal with changing development conditions, and ensure management are informed</p> <p>4.6 Manage bugs to ensure efficient bug handling and resolution</p> <p>4.7 Manage test environment, including setup, receipt of test builds and clean-up</p>
5. Finalise testing for release	<p>5.1 Produce testing results for management review prior to release</p> <p>5.2 Manage test product freeze for final release and final test run</p> <p>5.3 Confirm product, release enabling, and quality requirements have been met</p>

Required Skills and Knowledge

This section describes the skills and knowledge required for this unit.

Required skills

- analytical skills to define test plan details and identify bugs or issues
- communication skills to:
 - determine and define quality requirements statement
 - manage and coordinate testing processes and finalisation of testing
- literacy skills to produce technical documentation and reports
- planning skills to define test plan details
- technical skills to:
 - manage test environment
 - develop test plan details
 - select, install and configure test plan support software.

Required knowledge

- client requirements for platforms, hardware and software
- client system requirements, both functional and non-functional
- procedures for bug or issue management and identification
- test reporting requirements
- testing techniques, methods, test types, system dissection.

Evidence Guide

The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> • define quality requirements and an associated test plan for software • install and configure testing support software • manage testing process • identify bugs or issues accurately and concisely • finalise software testing process to enable product release.
Context of and specific resources for assessment	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> • system undergoing development with associated specifications • client requirements (verbal or written) for system quality requirements • testing support software • test environment • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	<p>A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:</p> <ul style="list-style-type: none"> • direct observation of candidate identifying or finding bugs and issues • review of reports prepared by candidate showing plans and management of testing.
Guidance information for assessment	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate.</p> <p>Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed.</p> <p>Indigenous people and other people from a non-English speaking background may need additional support.</p> <p>In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.</p>

Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

<i>Types of requirements</i> may include:	<ul style="list-style-type: none"> functional, such as audio and visual installation, such as new, upgrades, database and software non-functional, such as performance, load, stress and cultural platform or environment.
<i>Outstanding bugs</i> may include:	<ul style="list-style-type: none"> defined priorities with or without defined priorities.
<i>Limited release</i> may include:	<ul style="list-style-type: none"> alpha and beta release options environment-specific release internal or in-house release options external release options platform-specific release.
<i>Test cycles</i> may include:	<ul style="list-style-type: none"> delivery of testable product to client delivery of testable product to test team in-development tests, such as source-code updates or retrievals from source control, initial implementation complete time, pre-implementation time.
<i>Types of testing</i> may include:	<ul style="list-style-type: none"> acceptance data functionality functionality graphical user interface (GUI) load performance regression smoke stress unit.
<i>Testing methods</i> may include:	<ul style="list-style-type: none"> automated, such as coded testing or using automation-tool manual, such as exploratory or testing of specific test cases static and dynamic analysis.
<i>Testing technique</i> may include:	<ul style="list-style-type: none"> boundary value analysis equivalence way of designing test cases.
<i>Test-support software</i>	<ul style="list-style-type: none"> bug tracking software

may include:	<ul style="list-style-type: none"> • test automation software, such as: <ul style="list-style-type: none"> • WinRunner • LoadRunner • test case-management software, such as Test Director.
<i>Implementation details</i> and <i>reporting details</i> may include:	<ul style="list-style-type: none"> • communication and interaction levels between departments for cycles, such as development may or may not be approachable during particular processes or cycles • details of all processes and procedures • required documentation repositories, templates, structures and locations.
<i>Completeness techniques</i> may include:	<ul style="list-style-type: none"> • interdepartmental reviews • intradepartmental reviews • system dissection.

Unit Sector(s)

Game development