



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG509A Build using rapid application development

Release: 1

ICAPRG509A Build using rapid application development

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to build using rapid application development (RAD) tools.

Application of the Unit

This unit applies to programmers who are required to develop new systems.

RAD building for a software project should be carefully managed to deliver maximum benefit for cost and quality. Selection of the most appropriate RAD tool will take into consideration the targeted platform or multi-platform options.

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

1. Construct application using RAD	1.1 Capture and record business rules using a methodology well-suited for the chosen RAD and <i>RAD techniques</i> 1.2 Design <i>application</i> with focus on modularity and future extension 1.3 Develop code by other <i>RAD tools</i> 1.4 Build and demonstrate completed transaction to user for revision within the agreed terms of reference 1.5 Take action to ensure that design caters for continuous change by involving user in iteration process 1.6 Provide quality assurance (QA) testing throughout the phase and provide feedback to <i>appropriate person</i> 1.7 Demonstrate use of code optimisers and performance tools
2. Prepare handover stage	2.1 Track implemented modules and follow up where necessary with appropriate person 2.2 Review user and builds for each module in the deliverables as preparation for handover 2.3 Compare <i>specifications</i> and implementation schedules for each module and confirm the functional requirements according to specifications 2.4 Document the completed work

Required Skills and Knowledge

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

Required skills

- communication skills to liaise with users while managing iterative changes in design by involving end users in the development process
- literacy skills to write technical documents and present written material for review
- planning and organisational skills to keep accurate technical records
- problem-solving skills to solve problems with design
- technical skills to identify, analyse and evaluate a range of solutions using prototyping tools.

Required knowledge

- broad knowledge of industry-accepted prototyping tools
- overview knowledge of:
 - quality assurance practices
 - client business domain
 - role of stakeholders and the degree of stakeholder involvement
 - three or more current industry development methodologies
 - three or more programming languages.

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	Evidence of the ability to: • design application and develop high quality code according to client specifications • provide QA testing and monitoring • demonstrate use of code optimisers using RAD techniques • document completed work.
Context of and specific resources for assessment	Assessment must ensure access to: • CASE repository to facilitate the re-use of templates and components • CASE tools • code generator • detailed user requirements • prototyping software • requirements document, including model and scope • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit: • evaluation of completed code • review of documentation • verbal or written questioning to assess knowledge of prototyping and RAD process.
Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. 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. Indigenous people and other people from a non-English speaking background may need additional support. In cases where practical assessment is used it should be combined with targeted questioning to assess required

	knowledge.
--	------------

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>RAD techniques</i> may include:	• computer-aided software engineering (CASE) tools • iterative life cycles • prototyping • re-use of applications, templates and code • SWAT teams • timebox-development methodology • workshops.
<i>Application</i> may include:	• commercial software applications • organisation-specific software: • communication packages • database • graphic • presentation functionalities • spreadsheet • word-processing.
<i>RAD tools</i> may include:	• CASE tools • code generators.
<i>User</i> may include:	• department within the organisation • person within a department • third party.
<i>Appropriate person</i> may include:	• authorised business representative • client • supervisor.
<i>Specifications</i> may include:	• current system functionality • metrics • project plan • software requirements • technical requirements • user problem statement.

Unit Sector(s)

Programming and software development