Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG416A Manage a software component reuse library

Release: 1

ICAPRG416A Manage a software component reuse library

Modification History

Release	Comments
Release 1	This Unit first released with *ICA11 Information and Communications Technology Training Package version 1.0*

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to store, document and maintain IT components for reuse.

Application of the Unit

This unit applies to individuals in a software development area who are required to manage a library that contains reusable code.

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
Elements describe the essential outcomes of a unit of competency.	*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.*

Elements and Performance Criteria

1. Specify components for reuse library	1.1 Determine ***component*** suitability for reuse by checking its quality and uses 1.2 Review the size and complexity of ***reuse components*** for generalisation and remove ***project***-specific refinements 1.3 Evaluate components, including patterns, clusters and frameworks, and break down into smaller components for greater flexibility or use 1.4 Review components for duplication and remove duplicates
2. Document reuse library	2.1 Document components internally and externally to ensure efficient retrieval 2.2 Document public constants, data structures, component interfaces and limitations 2.3 Document possible relationships between data structures or objects 2.4 Provide example code, demonstrating the use of components within the documentation 2.5 Document ***development environment*** configuration 2.6 Provide simple test programs to prove the functionality of the library
3. Set up library structure	3.1 Classify data structures or objects in a consistent and logical manner 3.2 Ensure that library has a logical structure so that the ***user*** is aware of library contents and can understand the logic of how the functionality may be used by another developer 3.3 Develop the structure in a manner that avoids redundancy within the library 3.4 Configure development environment to automate build of reuse components and library 3.5 Take action to ensure interfaces to library components are consistent and abstracted 3.6 Add, update and remove reuse components as development is undertaken

Required Skills and Knowledge

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

Required skills

- analytical skills to:
  - classify data structures
  - determine the quality of reuse components for efficient retrieval and use
  - evaluate components to allow for greater flexibility or use
- communication skills to liaise with users
- literacy skills to provide technical documents
- planning and organisational skills to:
  - manage own work priorities
  - produce a logical library structure
- technical skills to:
  - provide example code to demonstrate functionality
  - use simple test programs to prove the functionality of the library.

Required knowledge

- abstraction of code module interfaces
- component libraries
- design paradigms
- patterns, frameworks and idioms
- reuse components and metrics
- reuse libraries content and structure.

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: • reuse libraries and manage components within the library structure • store, document and improve the quality of reuse components for efficient retrieval and use • organise and communicate library contents.
Context of and specific resources for assessment	Assessment must ensure access to: • software development environment • technical requirements • 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: • verbal or written questioning to assess candidate's knowledge of design paradigms • review of the candidate's library structure • evaluation of the library components added.
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.

Component may include:	• classes • code • design • documents • metadata • methods • modules • test cases.
Reuse components may include:	• code • design patterns • metadata • requirements • specifications.
Project may include:	• business improvement process • business involved in a total organisational change • ebusiness solution involving the total organisation or part of the organisation • systems-only change.
Development environment may include:	• computer language used • development methodology • development tools • operating systems • target environments • version control systems.
User may include:	• department within the organisation • person within a department • third party.

Unit Sector(s)

Programming and software development