



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG506A Design application architecture

Release: 1

ICAPRG506A Design application architecture

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 design the structure of software or systems components and how they interact. The unit focuses on layered architectural style.

Application of the Unit

This unit is relevant to those who work as software architects, developers, designers, software engineers or programmers responsible for designing and building solution architects.

This unit addresses the knowledge and processes necessary to produce the high-level design (blueprint) of a system.

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. Gather and confirm architecture requirements	1.1 Gather information regarding architectural requirements 1.2 Establish and document <i>architectural requirements</i>
2. Design layered architecture	2.1 Separate the areas of concern into logical <i>layers</i> 2.2 Determine the <i>cross-cutting concerns</i> 2.3 Define the system into components 2.4 Identify the responsibilities of each component 2.5 Identify interconnections between components
3. Plan a strategy to re-use components	3.1 Determine an appropriate strategy for communicating with external systems 3.2 Interact with existing legacy components
4. Design for globalisation and localisation	4.1 Determine culture-specific information 4.2 Consider database design features 4.3 Select appropriate user interface 4.4 Develop software product for worldwide distribution 4.5 Develop software product for specific country

Required Skills and Knowledge

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

Required skills

- analytical skills to:
 - assign a priority to each architectural requirement
 - determine areas of concern
- communication skills to:
 - interact with others of different cultures to globalise a software product
 - interact with stakeholders to gather architectural requirements
- initiative and enterprise skills to identify potential improvements to the structure of software and existing systems
- literacy skills to:
 - maintain a complete list of architectural requirements
 - prepare architectural requirements questionnaire to system stakeholders
- problem-solving skills to:
 - communicate with external systems
 - determine the possibility of reusing the existing legacy components
- technical skills to:
 - analyse software requirements
 - design software applications.

Required knowledge

- basic knowledge of database design
- current software development methodologies
- detailed knowledge of:
 - object-oriented programming
 - software development life cycle.

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"> • design layered applications to increase application performance, scalability and re-usability • create flexible and configurable applications and adapt to suit different locales • adapt a software product to a particular language and culture of the target market.
Context of and specific resources for assessment	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> • specific requirements, including functional and non-functional requirements • tools to design software architecture • culture-specific information for the target culture or language • 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"> • review of candidate's architecture requirements documentation • evaluation of software-layered architecture.
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.

<p><i>Architectural requirements</i> may include:</p>	<ul style="list-style-type: none"> • design requirement • functional requirements • non-functional requirements: <ul style="list-style-type: none"> • maintainability • performance • reliability • scalability • security • usability • quality requirements.
<p><i>Layers</i> may include:</p>	<ul style="list-style-type: none"> • business logic layer (BLL) • data-access layer (DAL) • services layer • user interface (UI) layer.
<p><i>Cross-cutting concerns</i> may include:</p>	<ul style="list-style-type: none"> • caching (improve performance) • communications (protocols) • configuration management • data protection • operational management • security: <ul style="list-style-type: none"> • authentication • authorisation.

Unit Sector(s)

Programming and software development