



**Australian Government**

**Department of Education, Employment and Workplace Relations**

# **ICAPRG510A Maintain custom software**

**Release: 1**

## ICAPRG510A Maintain custom software

### 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 maintain software so that it continues to meet client user requirements.

### Application of the Unit

This unit applies to programmers who are required to maintain existing software.

Organisations may use either software that has been developed in-house or off-the-shelf packages. Some customisation generally occurs with off-the-shelf packages. It is important that all changes made to standard software products are documented. Changes may be made in response to user requests or organisational requirements.

### 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

<b>Element</b>	<b>Performance Criteria</b>
<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. Determine software fault to be corrected	<p>1.1 Collect and review software fault details from sources</p> <p>1.2 Obtain <i>technical data</i> to assist in identifying problem</p> <p>1.3 Clarify nature of the problem with <i>appropriate person</i> where necessary</p>
2. Identify and isolate fault	<p>2.1 Review program documentation for specific modules in order to pinpoint problem areas</p> <p>2.2 Review source code for logic errors</p> <p>2.3 Read manuals, help files and 'read me files' to determine if there is a known fix</p> <p>2.4 Undertake additional testing to identify or duplicate fault</p> <p>2.5 Escalate difficult faults that cannot be identified</p>
3. Design fix for fault	<p>3.1 Ensure requirements to fix the fault are understood by client</p> <p>3.2 Consider alternative options and choose the most effective solution</p> <p>3.3 Consider the possible impact of the fix on other parts of the system</p> <p>3.4 Document changes according to organisational guidelines</p>
4. Carry out fix to software	<p>4.1 Identify and obtain access to appropriate software development tools, source code and libraries</p> <p>4.2 Construct appropriate code to correct the fault according to organisational and programming standards</p> <p>4.3 Compile or regenerate code for changed programs and associated modules</p> <p>4.4 Correct and resubmit code until error free</p> <p>4.5 Document changes according to organisational and programming standards</p>
5. Test fix and associated system areas	<p>5.1 Check logic to ensure that it works with test data, corrects original fault, and does not cause problems elsewhere</p> <p>5.2 Request users to perform acceptance testing and record outcomes</p>
6. Hand over to systems operations area	<p>6.1 Update <i>documentation</i> to reflect all changes made</p> <p>6.2 Confirm acceptance by systems operations and arrange for sign-off according to procedures</p>

## Required Skills and Knowledge

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

### Required skills

- analytical skills to review software fault details and interpret complex technical data
- communication skills to liaise with clients
- literacy skills to interpret and write technical documents
- planning and organisational skills to ensure adherence to standards and procedures in programming
- problem-solving skills to consider alternative options and possible impacts
- technical skills to use, customise and adapt software packages.

### Required knowledge

- detailed knowledge of:
  - concepts relating to system performance
  - concepts relating to testing of software systems
  - current industry-accepted hardware and software products, and their general features and capabilities
  - system's current functionality.

## 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.*

<b>Overview of assessment</b>	
<b>Critical aspects for assessment and evidence required to demonstrate competency in this unit</b>	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> <li>• effectively maintain custom software</li> <li>• apply a fix that works</li> <li>• deploy a possible range of solutions to produce the same results.</li> </ul>
<b>Context of and specific resources for assessment</b>	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> <li>• fault logs and help-desk reports</li> <li>• software development tools, documentation and environment</li> <li>• source code and libraries</li> <li>• appropriate learning and assessment support when required</li> <li>• modified equipment for people with special needs.</li> </ul>
<b>Method of assessment</b>	<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"> <li>• evaluation of completed code</li> <li>• review of candidate's updated program documentation</li> <li>• verbal or written questioning to determine candidate's understanding of:             <ul style="list-style-type: none"> <li>• impact of changes on applications and systems</li> <li>• programming standards.</li> </ul> </li> </ul>
<b>Guidance information for assessment</b>	<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.*

<b><i>Technical data</i></b> may be obtained from:	<ul style="list-style-type: none"> <li>• error messages</li> <li>• memory dumps</li> <li>• software traces</li> <li>• other information.</li> </ul>
<b><i>Appropriate person</i></b> may include:	<ul style="list-style-type: none"> <li>• authorised business representative</li> <li>• client</li> <li>• supervisor.</li> </ul>
<b><i>Documentation</i></b> may follow:	<ul style="list-style-type: none"> <li>• audit trails</li> <li>• International Organization for Standardization (ISO), International Electrotechnical Commission (IEC) and Australian Standards (AS) standards</li> <li>• maintaining equipment inventory, client training and satisfaction reports</li> <li>• naming standards</li> <li>• project-management templates and report writing</li> <li>• version control.</li> </ul>

## Unit Sector(s)

Programming and software development