



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG401A Maintain open-source code programs

Release: 1

ICAPRG401A Maintain open-source code programs

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 contribute as a member of an open-source software project community and to maintain open-source code. The open-source development model differs considerably from the traditional commercial model.

Application of the Unit

This unit applies to individuals working in software and system development and maintenance. They may work as open-source software developers who are required to build, test and distribute open-source software applications, or programmers responsible for integrating open-source components, tools or technologies into their applications.

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. Investigate open-source paradigm	1.1 Examine the open-source paradigm and demonstrate an understanding of the differences from the traditional software development models 1.2 Investigate and demonstrate understanding of the types of online resources 1.3 Investigate and demonstrate understanding of the types of project documentation 1.4 Recognise and demonstrate understanding of the role of an online community and international collaboration 1.5 Examine and demonstrate understanding of motivational factors for contributors to open-source code 1.6 Analyse and demonstrate understanding of open-source licensing models
2. Familiarise with target project	2.1 Examine online resources associated with the target project 2.2 Download pre-built executable binaries to install and run project 2.3 Download, read and demonstrate understanding of supporting documentation
3. Prepare for maintenance activities	3.1 Select and register with a relevant online community open-source group 3.2 Download nightly snapshots of latest source code and supporting documentation 3.3 Build and execute snapshot where appropriate
4. Maintain code	4.1 Access the project bug database and select bugs to be resolved or features to be added 4.2 Make changes to local copy of code to resolve selected bugs 4.3 Test resulting code to ensure it performs appropriately 4.4 Prepare code patch for submission 4.5 Submit code patch to project 4.6 Use appropriate software-development tools and environment
5. Maintain documentation	5.1 Access project documentation 5.2 Prepare and contribute new information or updates to existing documentation 5.3 Prepare and submit documentation changes to project
6. Participate in	6.1 Exchange messages with other project members and actively

community	participate in community activities 6.2 Take action to ensure exchanges are <i>socially acceptable</i> 6.3 Submit code and documentation code patches for inclusion 6.4 Access <i>online project</i> resources frequently to keep up-to-date with project and community developments 6.5 Take action to ensure <i>community-participation standards</i> are observed and maintained
-----------	---

Required Skills and Knowledge

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

Required skills

- analytical skills to:
 - apply some of the latest information technology tools and techniques
 - evaluate and compare a range of open-source tools and components to use in software development and maintenance
- communication skills to:
 - discuss issues with global peers
 - participate effectively in discussions on a number of information technology issues
 - participate in volatile communities of interest
- initiative and enterprise skills to:
 - apply information technology knowledge and skills in a multidisciplinary environment
 - participate in open-source projects
- literacy skills to:
 - read and interpret complex technical and non-technical information from a range of sources
 - write a clearly structured document or update existing documentation
- problem-solving skills to choose the best solution for a particular context
- research skills to identify and locate possible sources of required information for a target project
- technical skills to:
 - build existing software projects from source
 - create and maintain code
 - use basic language syntax and layout, data structures, arrays and standard algorithms
 - test and debug code at a high level.

Required knowledge

- detailed knowledge of:
 - debugging and testing techniques
 - documentation techniques, including internal documentation, user guides and technical documentation
 - open-source development methodology
- overview knowledge of online project communities.

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: • provide significant contribution to an open-source project in the form of: • program source code changes and/or additions • documentation changes or additions • online participation in the project direction • local proof of building of snapshots, code modifications and testing.
Context of and specific resources for assessment	Assessment must ensure access to: • software development environment • internet • technical requirements • databases • information repositories • programming languages • group-facilitation software • 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 open-source practices, open-source software and open-source issues such as distribution, compatibility and security • evaluation of candidate's: • participation • application code, test and documentation.
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.

Software may include:	• commercial software applications • customised software • in-house software • organisation-specific software • packaged software.
Online resources may include:	• community sites • CVS trees • FTP sites • gopher sites • news groups • online communities • websites.
Documentation may follow:	• audit trails • International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Australian Standards (AS) standards • naming standards • project management templates • report writing principles • version control.
Online community may include:	• local communities • online virtual communities of interest • organisational communities • main tools for online communities: • mailing lists • newsgroups • web conferencing.
Licensing models may include:	• Apache licences • Berkeley Software Distribution (BSD) • General Public License (GPL) • Lesser General Public License (LGPL) • Mozilla.

Software-development tools may include:	• computer language • development methodology • development tools • operating systems • target environments • version control systems.
Socially acceptable should be:	• designed and sent with regard to recipient's: • culture • ethnicity • location • race • social sensitivities.
Online project may include:	• commercial hosting facility hosting community code • CVS site • FTP site • other type of group repository • private website.
Community-participation standards may include:	• informal or formal rules and regulations used by groups to manage their online projects • procedures or guidelines • verbal or written policies.

Unit Sector(s)

Programming and software development