Secure Software Development Procedure

Section 1 - Purpose

(1) This Procedure defines the requirements for secure software development at the University.

Scope

(2) This Procedure applies to all Macquarie University (the University) Staff and Third-Parties involved in software development.

(3) This Procedure does not apply to:

a. Commercial of-the-Shelf (COTS) software that is acquired by the University (e.g., Software as a Service); and
b. Proof-of-Concept and (confirm third term) development activities.

Background

(4) The University is committed to maintaining a secure technology environment and as such, has established requirements to embed secure practices into the software development process. Establishing these requirements aims to decrease the likelihood of malicious exploitation of software.

Section 2 - Policy

(5) Refer to the Computer and Network Security Policy.

Section 3 - Procedures

General

(6) A secure software development lifecycle should be created by the software owner, to be maintained and regularly reviewed.

(7) Secure software development training should be included as part of the standard training for developers.

Development Tooling

(8) Software development activities should only be conducted using approved development tooling and environments.

(9) Open-source software used in development should be subject to a formal risk assessment before being approved for use within the University's environment.

Environment Segregation

(10) All software development should be conducted in accordance with the SecDevOps (Security-Development-Operations) standard.

(11) Artificial Intelligence (AI) tooling should not be given permission to deploy code directly into Production.

(12) Access to Development, Test and Production environments should be granted in accordance with the Computer and Network Security Policy.

Source Code Management

(13) Source code should be stored securely within an approved source code management tool.

(14) Access to the source code storage tools should be granted in accordance with the Computer and Network Security Policy.

Data Protection Requirements

(15) Development and quality assurance activities should not use data contained within Production databases/datastores.

(16) Test data that matches the structure of production data should be generated for all Development and Test environments.

(17) Developers should implement security controls based on the classification of the Information the system is expected to generate, store, process, and/or transmit in accordance with the Information Classification and Handling Procedure.

(18) All Confidential Information and above (per the Information Classification and Handling Procedure) should be encrypted at rest, in process, and in transit, in accordance with the Computer and Network Security Policy.

(19) Source code should not contain hardcoded secrets (e.g., credentials, tokens, API keys).

Third-Party Library Management

(20) Third-Party Libraries should be, at a minimum:

a. evaluated before use to ensure they are:
    i. regularly maintained/supported; and
    ii. provided by a trusted source.
b. approved for use;
c. regularly updated/kept up to date; and
d. scanned for vulnerabilities.

Outsourced Development

(21) Third-Party developers should provide developed software to the University for review and acceptance to ensure it meets the requirements specified in this Procedure, prior to deployment into Production.

(22) The development of software by Third-Parties should be governed by a contract or a Service Level Agreement (SLA) that includes minimum security requirements.

(23) Access to the University source code should be based on the principle of least privilege and restricted to those under either contract or an SLA.

Secure Software Development Lifecycle

(24) A secure software development lifecycle should be defined, which outlines the key phases for secure software

development, including:

a. preparation: processes and activities related to how the University intends to securely develop software, including approved tooling/coding languages and identifying risks;
b. design: formalisation of development objectives and requirements for the creation of software within development projects, including architecture design specifications, threat modelling and secure coding principles (e.g., least privilege, segregation of duties, validation of inputs/outputs);
c. development: definition of specific requirements when developing software components with the goal of reliably deploying software with minimum defects;
d. testing: processes and activities related to how the University confirms and tests developed software, including quality assurance and other review and evaluation activities;
e. deployment: requirements when deploying software; and
f. operations: activities necessary to ensure confidentiality, integrity, and availability of developed software is maintained throughout its lifetime.

Training and Awareness

(25) Secure software development training should include, but is not limited to, the following requirements:

a. use of Artificial Intelligence (AI) in development activities (e.g., GitHub Copilot);
b. secure coding practices (e.g., principle of least privilege, security by design, input validation);
c. assurance and testing (e.g., code audits, Security Vulnerability assessments, penetration testing and threat modelling);
d. cryptography (e.g., encryption, hashing, secure certificate management); and
e. secure coding practices/guidelines (e.g., Open Web Application Security Project).

Development Practices

(26) Development activities should follow the Open Web Application Security Project (OWASP) standards (where applicable), including but not limited to:

a. OWASP Top Ten/Top Ten for Mobile;
b. OWASP Testing Guide;
c. OWASP Application Security Verification Standard (ASVS);
d. OWASP Software Assurance Maturity Model (SAMM); and
e. OWASP Mobile Application Security Verification Standard (MASVS).

(27) Industry recognised secure coding guidelines should be followed for all programming languages used for software development within the University.

(28) Security requirements should be documented alongside the functional requirements of the software.

Assurance and Testing

(29) Security reviews must be planned and performed throughout all stages of software development, including:

a. code audits;
b. Security Vulnerability assessments;
c. penetration testing; and
d. threat modelling.

(30) Automated security testing tools (e.g., static application security testing, software composition analysis and dynamic application security testing), should be used to identify potential vulnerabilities.

(31) Acceptance testing programs and related criteria should be established for new systems or enhancements to existing systems.

Compliance and Exemptions

(32) Any exemption to this Procedure must be sought from the Chief Information Security Officer (CISO).

(33) Breaches of this Procedure by Staff will be managed in accordance with the applicable provisions of the Staff Code of Conduct and other relevant policy instruments.

Section 4 - Guidelines

(34) Nil.

Section 5 - Definitions

(35) The following definitions apply for the purpose of this Procedure:

a. Information means any information in either physical or electronic format that is generated, created, stored, purchased or received during the conduct of University operations.
b. Information Technology Resources, or IT Resources, includes, but is not limited to:
    i. All computers and all associated data networks and systems, internet access and network bandwidth, email, hardware, data storage, computer accounts, all OneID systems, media, software (both proprietary and those developed by the University) and telephony services.
    ii. Information Technology services provided jointly, or as part of a joint venture between the University and a research centre, school, institute affiliated with the University, a subsidiary organisation owned by the University or any other partner organisation.
    iii. Information Technology services provided by Third-Parties that have been engaged by the University.
c. Library in the context of Third-Party Library means a collection of pre-compiled, reusable code, routines, or modules developed and maintained outside the University that can be incorporated into software developed by or on behalf of the University, including open-source packages, frameworks, and software development kits (SDKs).
d. Security Vulnerability means a weakness in an IT Resource that can be exploited for malicious purposes.
e. Staff means an individual directly employed by the University and its controlled entities.
f. Third-Party means an individual or organisation working under contract with the University.

Status and Details

Status	Current
Effective Date	23rd July 2026
Review Date	23rd July 2031
Approval Authority	Deputy Vice-Chancellor (People and Operations)
Approval Date	23rd July 2026
Expiry Date	Not Applicable
Responsible Executive	Eric Knight Deputy Vice-Chancellor (People and Operations)
Responsible Officer	Jonathan Covell Chief Information and Digital Officer
Enquiries Contact	Andrew Karvinen Chief Information Security Officer