Category	IT Infrastructure and Services	**Policy Number**	SUP-ITS-401	
Classification	Public	**Version**	1.0	
Responsible Office	CAFO	**Policy Owner**	ITD Director	
Date Approved	26 March 2025	**Effective Date**	1 May 2025	
Date Last Reviewed	New Policy	**Due Date for Next Review**	26 March 2028	

POLICY
Software Application Development

1. Purpose

1.1 The purpose of this policy is to outline the software application development guidelines governing the software development lifecycle at Zayed University.

1.2 The policy ensures adherence to software development industry best practices, and protection of Zayed University software Intellectual Property.

2. Scope of Application

This policy applies to all Zayed University software development activities. Software development activities include designing, coding, testing, releasing, storing, archiving, and providing access to software applications.

3. Definitions

CAFO	Chief Administration and Finance Officer
IP	Intellectual Property
ITD	Information Technology Department
SDLC	Software Development Life Cycle
University	Zayed University

4. Policy

4.1 All software developed by Zayed University ("**University**") development employees, contractors, and vendors will be owned by the University as Zayed University Intellectual Property ("**IP**"). Any use and/or duplication of IP developed software must be authorized by the University.

4.2 All tools, components, and platforms used to develop University software must be legally licensed for software development, deployment, and use by the University.

4.3 Software development activities must be governed by procedures and controls established and managed by the University Information Technology Department ("**ITD**").

4.4 The procedures and controls will govern the design, development, transmission, copy, transfer, testing, release, access, and support of software solutions on secure, authorized terminals and devices.

4.5 The procedures and controls will support and provide oversight to the procurement of software development services at the University.

4.6 The procedures and controls must align with software development industry guidelines, norms, and best practices ensuring software development efficiency, quality, and security.

4.7 The procedures and controls will ensure that adequate infrastructure, capacity, and tools are allocated for high quality, efficient, and secure software development activities at the University.

4.8 The procedures and controls will ensure adequate storage, authorized access, transmission, source code control, and backup of all software developed by the University.

5. Guidelines and Controls

5.1 Software Development

5.1.1 The developed software code must use uniform technology platforms.

5.1.2 The code must be readable, traceable, auditable, and secure.

5.1.3 The code must be reusable, increasing usability efficiency.

5.1.4 The code must use distributed development platforms supporting code sharing, team development, and peer validation.

5.1.5 The code must be tested and validated using tools, and must be peer reviewed prior to release.

5.1.6 Coding activities must be logged. All codes must be checked-in regularly to a central code repository maintaining coding history, updates, and bug fixes.

5.1.7 Code storage, transmission, and sharing must be through University-authorized channels and to University-authorized individuals.

5.1.8 The code, specifically the source code, will be the intellectual property of the University.

5.1.9 The source code must be safeguarded to avoid leakage, duplication, or repurpose by unauthorized individuals or organizations.

5.1.10 All source code transmission, duplication, and sharing must be authorized by the ITD Director and logged appropriately. The code, specifically the source code, must only be transmitted using secured mediums.

5.1.11 The code, specifically the source code, may only be shared after authorization through documented contracts and non-disclosure agreements protecting University IP have been obtained.

5.2 Software Development Environments

5.2.1 Software development, testing, and user acceptance environments must be independent from live operational environments.

5.2.2 Software development activities of coding, testing, validation, and acceptance must be conducted in non-operational environments.

5.2.3 Exceptional approval must be obtained from the ITD Director for software development and update activities made directly on operational environments. Such actions must be undertaken in controlled situations that safeguard against the loss of data and services.

5.2.4 The same level of data protection must be applied to operational data that is replicated to non-operational environments. If this is not possible, sensitive operational data must either be removed or scrambled before the data is replicated.

5.2.5 Operational environment replication must follow the pre-defined ITD procedures and go through the ITD change management process.

5.2.6 All development environments must remain secure with access given to authorized users only.

5.3 Software Development Tools

5.3.1 The software development tools used must support cloud and non-cloud architectures, provide agility and delivery cadence, maintain security and compliance automation, ensure visibility, traceability and auditability of the software development activities and code developed within the University.

5.3.2 The software development tools must provide the most appropriate platform in terms of functionality, security, and performance.

5.3.3 The software development tools must include code repository and code version control tools.

5.3.4 All software development tools must be licensed and authorized for use.

5.3.5 Any proposed use of open-source software must be submitted to and approved by ITD before being used.

5.4 Secure Coding

5.4.1 Software codes need to be secure by design. Credential storing, password hardcoding, non-secure connection usage, storage of data on unsanctioned devices, and unsecure caching of data must be avoided.

5.4.2 All data need to be stored in secure data repositories, databases, and other environments in order to maintain the data confidentiality, integrity, and availability in accordance with the University's data classification and handling procedures.

5.4.3 If non-secure coding practices cannot be avoided, the coding shortcomings must be managed through secure sandboxing and other environmental safeguards.

5.4.4 All software developed by University developers, contractors, and vendors needs to pass through security testing using security testing tools before release to operational environments. Shortcomings and issues need to be highlighted, fixed, and registered as risks before being released and/or accepted by domain owners.

5.4.5 All non-secure coding components that are to be released to an operational environment must be identified as risks, and authorized by the ITD Director before being released.

5.5 Software Development and Release Management

5.5.1 Software development activities need to be pre-authorized by the ITD Director and tracked and accounted for by the appropriate budget controllers.

5.5.2 All resources, tools, and labor costs need to be recorded and accumulated as software development cost.

5.5.3 Software that has been developed needs to go through defined regression, unit, functional, and security testing before being authorized by the appropriate budget controllers for User Acceptance Testing.

5.5.4 Software that has passed the User Acceptance Testing needs to go through the ITD Change Management process for deployment in the University operational environment.

5.5.5 All software development, testing, and release activities that bypass the above controls needs to be authorized by the ITD Director.

5.6 Risk Management

5.6.1 Potential risks, especially those focusing on areas such as data security and regulatory compliance, must be identified and documented in line with the University's risk management strategies.

5.6.2 Each phase of software development must be aligned with the University's overall risk management policies to ensure regulatory compliance, data protection, and incident response preparedness.

5.6.3 Risks must be tracked and mitigated throughout the Software Development Life Cycle ("**SDLC**"), in line with the established procedures.

5.6.4 Development teams must be provided with ongoing training on risk assessment, mitigation strategies, and compliance best practices.

5.7 Intellectual Property Compliance

5.7.1. The University's IP ownership must be prioritized, differentiating between University-owned software, open-source code, and third-party integrations.

5.7.2 Documentation for third-party software applications must be provided by the vendors as per the licensing agreement.

5.7.3 IP audits on software projects must be conducted to ensure ongoing compliance with licensing terms and University policies.

6. Related Policies and Laws

- SUP-ITS-201 Access Control
- SUP-ITS-203 Information Security
- SUP-ITS-204 Data Security

7. Administration

This policy is administered by the Operating Systems unit, ITD.

8. Revision History

Date	Revision	Ver.
1 May 2025	Chair of the Board's Decision issued (CBD#10 of 2025).	1.0
26 March 2025	Approved by FAIAC.	
11 March 2025	Endorsed by UPSC.	
6 November 2024	In response to FAIAC feedback:	

	• Added Articles 5.6 and 5.7.	
13 June 2024	Reviewed by FAIAC and returned to ITD with feedback.	
22 June 2023	Endorsed by the Executive Committee.	
6 October 2022	Reworded for greater clarification.	
6 July 2022	Re-reviewed by CAFO Management Council based on Transformation Office advisor's recommendation and endorsed for further approval.	
3 June 2022	Benchmarked and reformatted as policy encompassing procedural elements as governing policy directives.	
10 June 2021	Reformatted and edited, with relevant information added from the procedures, by the VPO.	
28 April 2021	New policy reviewed by CAFO Management Council and endorsed for further approval.	