THE UNITED REPUBLIC OF TANZANIA

PRESIDENT'S OFFICE

e-GOVERNMENT AUTHORITY

Document Title

Standards and Guidelines for Development, Acquisition, Operation and Maintenance of e-Government Applications

Document Number

eGA/EXT/APA/005

Approval	Name	Job Title/ Role	Signature	Date
Approved by	Dr. Mussa M. Kissaka	Board Chairperson		18/02/2026

Version: 3.0 – February 2026

PREFACE

In the last few decades, the use of ICT as enabler for improving Government operations and service delivery to citizens has not only become rampant, but also inevitable prerequisite for enhancing its efficient and effective. In the quest of reaping the benefits brought about by the use of ICT, public institutions in Tanzania have vigorously been striving to take its advantage but in an uncontrolled manner that resulted into emergence of a number of challenges relating to duplication of efforts, silo initiatives, high cost and security vulnerabilities.

However, if ICT is appropriately used by public institutions, it would effectively contribute to the improvement of their internal operations as well as public service delivery, which are expedient, ease to access and affordable. Therefore, in order to achieve these objectives, it was apparent for enactment of the e-Government Act No.10 of 2019 and its Regulations that was declared through a Government Notice No. 75 of February 7, 2020, which provide guidance on proper approach for implementing e-Government and establishment of e-Government Authority with mandate of coordinating, promoting and overseeing e-government implementations as well as enforcing compliance with laws, regulations, standards and guidelines related to e-government implementations in public institutions.

In this context, Section 5 (2) (c) and 22 (3) of the Act requires and empowers e-Government Authority to ensure that public institutions are implementing e-government initiatives in a manner that ensures the anticipated benefits are achieved. Pursuant to these provisions, the Authority has prepared this document to prescribe standards for Development, Acquisition, Operations and Maintenance of e-Government Applications.

Therefore, we call for all public institutions to effectively observe these standards when embarking on an e-government initiative relating to application development, acquisition, operations and maintenance.

Dr. Mussa M. Kissaka
BOARD CHAIRPERSON

Table of Contents

ACRONYM

Application Computer programs, procedures, rules and associated documentation and data pertaining to the operation of a computer system

Application Acquisition A process that is intended to assist public institutions with the selection, purchase and, if applicable, implementation of applications, frameworks or other software components.

Application Development A process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks or other software components.

Application or software development life cycle A systematic approach to the creation of software or application. This life cycle typically includes a requirements, analysis, design, development, test, implementation and post-implementation phases

Audit or review An independent review for the purpose of assessing compliance with software requirements, specification, baseline, standards, procedures, instructions, development, and other requirements

Baseline A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for father development and that can be changed only through change management procedures

Business Support Applications These are applications used by user departments to facilitate performance of business support functions, such as human resources and administration, customer management, sales, marketing, finance and accounting, audit, planning, performance management, procurement, fleet, assets, projects, office communications, file and records management etc.

Core Business Applications These are applications used by user departments to perform main or mandated business functions of their respective institutions. These applications are usually developed (custom-made) in order to meet specific needs of the respective business.

Evaluation A technique in which requirements, design, development and test results are examined in detail by a person or group to detect problems. The results are documented

Government Data Center Centralized facilities that offer hosting/co-location services to public Institutions.

ICT Support Applications These are technical applications that are necessary to support the business support and core business applications. They include but are not limited to systems/networks monitoring and management systems, active directory, security systems, etc.

Maintenance To repair, change, or add to software product

Off-the-shelf Applications Are readymade software available for use with necessary customization in order to satisfy the needs of the respective public institution. They include Commercial Off-the-Shelf (COTS), Modifiable Off-the-Shelf (MOTS), Government Off-the-Shelf (GOTS), and Niche Off-the-Shelf (NOTS).

Outsourced Application Development A practice of hiring a third-party programmer/company to offer services related to specific activities or all activities related to application development.

Platform Dependence Refers to applications that run under only one operating environment. For example, Windows running on x86 hardware or Solaris running on SPARC hardware.

Quality Application Applications that meet specified requirements and/or user/customer needs and expectations.

Software as a Service (SaaS) A software distribution model in which a third-party provider hosts application and makes them available to customers over the internet or a private network.

System Custodian A person who has a responsibility for taking care of or protecting systems/applications, normally the head of ICT department/unit. The system custodian is a key contributor in developing system design and security specifications to ensure that they are documented, tested, and implemented.

System Owner A person who owns a business process, who is a key contributor in developing business requirement specifications to ensure that the business operational needs are met.

GLOSSARY

CI/CD	Continuous Integration and Continuous Deployment
DevSecOps	Development, Security and Operations
e-GA	e-Government Authority
ERP	Enterprises Resource Planning
ICT	Information & Communication Technology
OLA	Operation Level Agreement
RFB	Request for Bid
RFI	Request for Information
RFP	Request for Proposal
SaaS	Software as a Service
SCM	Source Code Management
SDD	System Design Document
SDLC	Software Development Life Cycle Methodology
SLA	Service Level Agreements
SRS	Systems Requirement Specification

1. INTRODUCTION

1.1. Overview

e-Government Authority "e-GA" is a public institution established by e-Government Act No.10 of 2019 with mandate to coordinate, oversee, and promote e-Government initiatives and enforce e-Government related policies, laws, regulations, standards and guidelines to Public Institutions.

This document establishes standards and guidelines to be adhered by public institutions during application development, acquisition maintenance and operations in line with the section 5(2) (i) of e-Government Act No.10 of 2019.

1.2. Purpose

This document defines the standards and guidelines to be applied when developing, acquiring, operating, and maintaining applications within Public Institutions. It focuses on outlining the specific standards and guidelines that must be followed during the application lifecycle. The document is intended for application developers responsible for designing, developing, and maintaining applications for their institutions, including external contractors, consultants, and business partners.

1.3. Rationale

e-Government implementation, including application development, acquisition, operations, and maintenance has been carried out through uncoordinated approaches, resulting in duplicated efforts, siloed initiatives, a lack of system integration, increased acquisition and operational costs, and heightened security concerns. To address these challenges, e-GA developed this document to provide a unified framework of standards and guidelines that will ensure consistency, interoperability, cost-effectiveness, and improved security in the development, acquisition, operation, and maintenance of e-Government applications.

1.4. Scope

This document will be used by all public institutions during development, acquisition, operation and maintenance of e-Government applications.

2. STANDARDS AND GUIDELINES

2.1. THE STANDARDS

2.1.1. Application Development Standards

2.1.1.1. General Standards

i. Issue a written statement establishing and/or selecting a Hybrid which includes Waterfall and Agile software development life cycle methodology (SDLC) as a means for constructing and managing the process of developing application software and use it for the duration of the entire development.

ii. If required to change the selected application development methodology, a change request must be issued in accordance to change management process.

iii. The application development project must operate in compliance with Enterprise Architecture Standards, Principles, Processes, and Methods.

iv. The application development process shall adopt the DevSecOps approach.

v. Evaluate the architecture of the application, interface and database design. The results of the evaluations must be documented. Below are the criteria to be considered:

a. Traceability to the requirements of the application;

b. External consistency with requirement of the application;

c. Internal consistency between the application components;

d. Appropriateness of design methods and standards used;

e. Feasibility of detailed design; and

f. Feasibility of operation and maintenance.

2.1.1.2. Requirement Gathering Standards

i. Business and system requirements must be gathered. Activities in this area include:

a. Review existing systems/process;

b. Describe data/system/process;

c. Perform business process mapping;

d. Identify problem, areas/opportunities;

   e. Identify user needs/wants;
   f. Conduct interviews;
   g. Identify manual and automated processes;
   h. Draw conceptual flow;
   i. Identify follow on projects/phases;
   j. Identify inputs (functional description);
   k. Data entry screens; and
   l. Inputs from outside sources.

ii. Establish and document business/application requirement. In order to achieve this, following items shall be followed:

   a. Document requirements;
   b. Document assumptions;
   c. Document outstanding issues;
   d. Estimate data storage requirements;
   e. Identify legislative/contractual/security/privacy/access requirement;
   f. Document reporting requirements;
   g. Establish training requirements;
   h. Conduct initial walkthrough; and
   i. Obtain sign-off and approval.

iii. Provide the following minimum set of documentation as part of requirement gathering standards:

   a. Business requirement document; and
   b. Systems Requirements Specifications.

2.1.1.3. Requirements Analysis Standards

i. Analyze Business and system requirement.

ii. The specific intended use of the system to be developed must be analyzed to specify system requirements. The system requirements specification shall describe:

   a. Functions and capabilities of the system;
   b. Business, organizational and user requirements;
   c. Safety, security, information, privacy, interface, operations, and maintenance requirements; and

   d. Design constraints and qualification requirements. The system requirement specification must be documented.

iii. The system requirements must be evaluated based on the criteria on 2.1.1.1 (v). The results of evaluations must be documented.

iv. Establish and document application requirements, including the quality characteristics specifications, described below:
   a. Functional and capability specifications, including performance, physical characteristic, and environmental conditions under which the application is to perform;
   b. Interface external to the application module/component/service;
   c. Testing requirement;
   d. Privacy and security specifications, including those related to compromise of sensitive information;
   e. Data definition and database requirements;
   f. Installation and acceptance requirement of the delivered application product of the operation and maintenance site(s);
   g. User documentation;
   h. User operation and execution requirements; and
   i. User maintenance requirements.

v. Provide the following minimum set of documentation as part of requirement analysis standards:
   a. systems analysis document;
   b. application requirements and specification;
   c. interface requirement/specification;
   d. operational/support requirement;
   e. System/subsystem specification.
   f. Software requirement specification;
   g. Analysis class;
   h. Use-case model;
   i. Use-case package; and
   j. User-interface prototype.

vi. Upon successful completion of the review(s), a baseline for the requirements of the application must be established and formal sign off must be obtained.

2.1.1.4. Design Standards

i. Perform activities/tasks related to design. Activities include:

   a. Design system flow;

   b. Develop data model, Create physical data model;

   c. Develop data dictionary;

   d. Design screens, screen navigation, data entry screens, inquiry screens, help screens, online documentation, Design reports, Forms, Report distribution system, User generated reports, Design Patterns, Existing system modifications;

   e. Conduct design walkthrough;

   f. Conceptual flow/procedures, and

   g. Process Implementation.

ii. A top-level architecture of the system must be established and documented. The architecture shall identify items of hardware, application/software, and manual operations. It shall be ensured that all the system requirements are allocated among the items. Hardware configuration items, application/software configuration items, and manual operations shall be subsequently identified from those items.

iii. Evaluate system architecture and requirements for the application/software based on the criteria on 2.1.1.1 (v). The results of the evaluations must be documented.

iv. Transform the requirement for the application into an architecture that describe its top-level structure and identifies the application components. It must be ensured that all the requirements for the application are allocated to its application components and further refined to facilitate detailed design. The architecture of the application must be documented.

v. Develop and document a top-level design for the interface external to the application and between the application components of the application.

vi. Develop and document a top-level design for the database.

vii. Develop and document preliminary versions of user documentation.

viii. Define and document preliminary test requirements and the schedule for application integration.

ix. Evaluate the architecture of the application, interface and database designs based on the criteria on 2.1.1.1 (v). The results of the evaluations must be documented.

x. Develop a detailed design for each application module/component/service of the application. These shall be refined into lower levels containing application units that can be coded, compiled, and tested. It shall be ensured that all the application requirements are allocated for the application components to application units. The detailed design must be documented.

xi. Update the user documentation as necessary such as end-user documentation (tutorials and user manuals) and system administrator documentation (troubleshooting, installation, and administration manuals).

xii. Evaluate the application detailed design and test requirements based on the criteria on 2.1.1.1 (v). The results of the evaluations must be documented.

xiii. Provide the following minimum set of documentation as part of design standards:

a. System Design Document (SDD);

b. Database design;

2.1.1.5. Development Standards

i. Review and analyze architecture/design documentation and construct/code to design specification.

ii. Develop application that can use different types of databases and can be supported by different operating systems.

iii. Perform the activities/tasks related to construction or coding. Activities/Tasks include:

a. Construct the application, components, services including Data entry screens, Inquiry screens, Menu screens, Online help screens, Batch programs, Changes to existing programs, Conversion programs, Build and load files/tables, Build job streams;

b. Develop test cases, Unit test programs, Develop secure code; and

    c. Develop application/software documentation, Users guide, Turnover documentation, Training materials, Conduct initial turnover walkthrough, Schedule turnover dates.

iv. Develop and document the following, application unit, database, test procedures and data for testing each application unit and database.

v. Define and document test requirements and schedule for testing application units. The test requirements shall include stressing the application unit at the limits of its requirements.

vi. Update the test requirement and the schedule for application integration.

vii. Test each application unit and database ensuring that it satisfies its requirements. The test results must be documented.

viii. Evaluate application code and test results based on the criteria on 2.1.1.1 (v). The results of the evaluations must be documented.

ix. Develop an integration plan to integrate the application units and application components into the application. The plan shall include test requirements, procedures, data, responsibilities, and schedule. The plan must be documented.

x. Integrate the application units and application components and test as the aggregates are developed in accordance with the integration plan. It shall be ensured that each aggregate satisfies the requirements of the application and that the application is integrated at the conclusion of the integration activity. The integration and test results must be documented.

xi. Develop and document, for each requirement of the application, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting Application Testing. Ensure that the integrated application is ready for Application Testing.

xii. Evaluate the integration plan, design, code, tests, test results, and user documentation based on the criteria on 2.1.1.1 (v). considering the criteria listed below. The results of the evaluations must be documented.

xiii. Upon successful completion of the review(s), a baseline for the construction of the application must be established and formal sign-off must be obtained.

xiv. Ensure storage of source code follows reasonable security protocols to ensure secure access and accountability for use of the source code.

2.1.1.6. Testing Standards

i. Conduct testing in accordance with the requirements for the application. It must be ensured that the implementation of each application requirement is tested for compliance. The testing results must be documented. Activities/Tasks include:

   a. Application test, software test, run parallel test and document results;

   b. User acceptance test, security test, develop test procedures and document results; and

   c. Issue production readiness recommendation.

ii. Evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations must be documented:

   a. Test coverage of the requirements of the application;

   b. Conformance to expected results;

   c. Feasibility of system integration and testing, if conducted; and

   d. Feasibility of operation and maintenance.

iii. Support the user testing. The results of the testing must be documented.

iv. Upon successful completion of the test:

   a. Update and prepare the deliverable application product for system integration, system testing, application installation, or application acceptance support as applicable; and

   b. Establish a baseline for the design and code of the application.

v. System Integration Testing

   i. The application must be integrated, with hardware configuration items, manual operations, and other systems as necessary, into the system. The aggregates must be tested, as they are developed, against their requirements. The integration and the test results must be documented.

   ii. For each requirement of the system, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting System Testing must be developed and documented. Ensure that the integrated system is ready for System Testing.

iii. The integrated system must be evaluated considering the criteria listed below. The results of the evaluations must be documented.

a. Test coverage of system requirements;

b. Appropriateness of test methods and standards used;

c. Conformance to expected results;

d. Feasibility of system testing; and

e. Feasibility of operation and maintenance;

iv. System testing must be conducted in accordance with the requirements specified for the system. It shall be ensured that the implementation of each system requirement is tested for compliance and that the system is ready for delivery. The testing results must be documented;

v. Upon successful completion of the system testing, if conducted:

a. Update and prepare the deliverable application product for Application Installation and Application Acceptance Support; and

b. Establish a baseline for the design and code of the application.

vi. Provide the following minimum set of documentation as part of testing standards:

a. User Acceptance Testing Report; and

b. System Acceptance Testing Report.

vii. Upon successful completion of the review(s), a baseline for the testing of the application must be established and formal sign-off must be obtained.

2.1.1.7. Implementation Standards

i. Develop a plan to install the application product in the target environment as designated. The resources and information necessary to install the application product shall be determined and be available. The project team shall assist the acquirer with the set-up activities. Where the installed application product is replacing an existing system, support any parallel running activities that are required. The installation plan must be documented.

ii. Install the application product in accordance with the installation plan. It shall be ensured that the application code and databases initialize, execute, and terminate. The installation events and results must be documented.

iii. Support acceptance review and testing of the application product. Acceptance review and testing shall consider the results of the Reviews, Audits, Application Testing, Security Testing, and System Testing. The results of the acceptance review and testing must be documented.

iv. Complete and deliver the application product.

v. Provide initial and continuing training and support as outlined in the implementation plan.

vi. Provide the following minimum set of documentation as part of implementation standards:

a. Application user manual; and

b. Application operator manual.

vii. Upon successful completion of the review(s), a baseline for the implementation of the application must be established and formal sign-off must be obtained.

2.1.1.8. Post Implementation Standards

i. Provide an integral part of the activities, a plan for a post-implementation review of the application development project.

ii. Perform a post implementation review to support continuous improvement of the implemented application system and to assess whether:

a. The application objectives have been achieved;

b. The application meets users' needs;

c. The project was delivered within the approved budget and schedule; and

d. Any variances in cost or time are identified, analyzed, and documented to inform future projects and improve project planning and governance.

iii. Results or report of a post-implementation review of the application development system be conducted, documented and retained for periodic reviews.

2.1.2. Application Acquisition Standards

The term acquisition refers to all stages from buying, introducing, applying, adopting, adapting and developing. The need to acquire an application is derived from various reasons like large variety of ICT applications, rapid change in new technology and involvement of several entities in the organization.

Public Institution may acquire an application by buying, leasing, outsourcing or any combination. For the Public Institution, before making the decision to acquire an application, the detailed requirements must be clearly identified along with the organization objectives whether building, leasing or buying the application shall consider a value-risk matrix to determine which options can be applied. The application acquisition shall involve the identification and analysis of alternative solutions that are each compared with the established public institution requirements.

A public institution must adhere to the following standards when making the decision to acquire an application;

2.1.2.1. Identifying, Planning and Justifying the Application Requirements

i. Identify the business problem to be addressed and the institutional objectives before planning or procuring an application.

ii. Define and document application requirements that specify the application objectives, scope, and expected outcomes.

iii. Gather application requirements using appropriate methods, including interviews, questionnaires, analysis of existing systems, and benchmarking with similar applications.

iv. Conclude the application requirements stage with a justified decision on the selected application, including the implementation timetable, budget, and agreed application expectations.

2.1.2.2. Restructuring Information System Architecture

i. Following a thorough application analysis, review and restructure the information system architecture.

ii. The information system architecture design must articulate the flow of information, data hierarchy, application functionality, and technical feasibility.

iii. The outcome of this stage is a strategic-level plan that directs the acquisition of applications in accordance with the defined information system architecture constraints.

2.1.2.3. Identifying a Development Alternative

i. Identify and select an appropriate application development or acquisition method after completing the information system architecture design.

ii. Application development or acquisition methods may include in-house development, off-the-shelf solutions, custom made system for a vendor, software leasing from application service providers or leasing through utility computing (contracted development) and outsourcing a system from other institutions.

iii. During the selection process, management must evaluate the advantages and disadvantages of each procurement option.

iv. The selected option must be best-fit with institution business plan that has been documented.

v. The Application development or acquisition initiatives, whether developed internally or sourced externally, must support the institution's business strategy and ICT strategy.

2.1.2.4. Conducting a Feasibility Analysis

i. As part of the assessment for acquiring application solutions, the institution conducts a feasibility analysis for each proposed alternative.

ii. The feasibility analysis is performed in accordance with the applicable Standards and Guidelines for Government ICT Project Implementation.

iii. The feasibility analysis evaluates constraints associated with each alternative from both technical and business perspectives.

iv. A feasibility report is prepared to document the findings and support informed decision-making.

v. Upon completion of the feasibility analysis, a risk analysis review is conducted to assess the security of the proposed application, including potential threats, vulnerabilities, impacts, and the feasibility of controls to mitigate identified risks.

vi. The feasibility analysis incorporates the following evaluation categories:

a. Economic Feasibility

Evaluates the cost–benefit justification of the proposed application by assessing all relevant costs, including procurement, implementation, operation, maintenance, training, infrastructure, and support. This evaluation ensures that the proposed solution is affordable, remains within approved budget limits, and promotes efficient use of resources.

b. Technical Feasibility

Assesses the technical suitability of the proposed application by evaluating the availability and adequacy of the institution's infrastructure, hardware, software, network capacity, and technical skills. This evaluation also considers compatibility with existing systems and the ability to support reliability, scalability, and future growth.

c. Operational Feasibility

Examines the extent to which the proposed application can be effectively integrated into existing business operations. This evaluation considers required organizational and process changes and ensures that the application addresses identified business problems and enhances operational effectiveness.

d. Legal and Contractual Feasibility

Evaluates compliance with applicable laws, regulations, and contractual obligations, including public procurement requirements. This assessment ensures that the acquisition and use of the proposed application do not introduce legal or regulatory risks.

2.1.2.5. Performing the Selection Procedure

i. Requests proposals from prospective providers, evaluates the submissions received, and selects the most suitable alternative.

ii. Request the proposal from providers by using various methods, including:

a. Request for Information (RFI), used to obtain information from vendors to identify potential solutions and viable alternatives that may address institutional needs.

b. Request for Bid (RFB), used to procure specific items or services when vendors are able to meet defined technical and functional specifications, or when a single vendor is capable of meeting those requirements.

c. Request for Proposal (RFP), used to define the minimum acceptable functional, technical, and contractual requirements and may lead to a procurement decision or further negotiations with vendors.

2.1.2.6. Proposal Evaluation Procedure

i. Review submitted proposals using a list of objective selection criteria and decide the best match between the product features and functionality with the identified requirements.

ii. Follow these six steps in selecting a software vendor with their application package:

a. Examining potential vendors background

Identify potential software providers using sources such as software catalogs, vendor lists from hardware suppliers, technical and trade journals, experienced consultants, and web-based searches.

Apply preliminary evaluation criteria to screen and eliminate unqualified vendors based on factors such as track record, reputation, and prior performance feedback.

b. Determine the evaluation criteria

Define detailed evaluation criteria for selecting the most suitable software vendor and application package, using vendor responses to the Request for Proposal (RFP).

Evaluate proposals against criteria that include vendor characteristics, functional and technical requirements, total project cost, solution scalability, project timeline, quality of documentation, and the vendor's support and maintenance offerings.

c. Evaluating providers and their applications

Assess gaps between institutional requirements and the capabilities of the vendors and their proposed application packages.

d. Selecting the provider and its solution

Select the software vendor and application package based on the nature of the application and its alignment with institutional requirements.

Conduct negotiations with shortlisted vendors to determine how proposed solutions can be configured or modified to address identified gaps. Additionally, consider feedbacks from users who will work with the system and the ICT staff who will support the system into the final selection decision.

Apply defined selection criteria to identify the software application package that best meets institutional needs and requirements, including:

- usability and functionality;
- cost-benefit analysis;

- upgrade policy and cost;
- vendor reputation;
- system flexibility and scalability;
- manageability;
- quality of documentation;
- hardware and networking resources;
- upgradeability;
- required training;
- system security;
- maintenance and operational requirement;
- user easiness to learn;
- performance measurement;
- interoperability and data handling;
- ease of integration;
- reliability measurement; and
- Compatibility with other application.

e. Negotiate a contract

Initiate contract negotiations with the selected vendor to agree on software pricing, licensing terms, and the scope of vendor support services.

Document contractual terms to include detailed system specifications, services to be provided by the vendor, support and maintenance arrangements, and all other relevant contractual conditions.

f. Establishing a service level agreement (SLA)

Establish a Service Level Agreement (SLA) with the vendor to define roles and responsibilities, service scope, performance objectives, quality metrics, and escalation or contingency scenarios.

Ensure the SLA provides a clear framework for delivering support services and preserves the institution's control, rights, and oversight over its systems.

2.1.2.7. Implementing the Selected Solution

Agree on an acceptance and implementation plan with the vendor upon completion of contract negotiations to prepare the selected application for installation or development.

2.1.3. Application Operations and Maintenance Standards

2.1.3.1. Application Operations

i. Application Availability and Performance

a. Monitor application availability, performance, and capacity against defined service levels.

b. Implement performance baselines and thresholds to detect and address degradation proactively.

c. Conduct regular capacity planning to support current and future demand.

ii. Incident and Problem Management

a. Detect, log, classify, and resolve application incidents in a timely manner.

b. Perform root cause analysis for recurring or high-impact incidents.

c. Maintain a knowledge base to support faster incident resolution.

2.1.3.2. System Maintenance

System maintenance encompasses various activities arising from problem reports, enhancement requests, or ad hoc requests, which may occur concurrently throughout the maintenance cycle.

The system maintenance cycle consists of the following four stages:

i. Stage 1 – Initiation;
ii. Stage 2 – Impact analysis;
iii. Stage 3 – Disposition; and
iv. Stage 4 – Implementation.

i. System Maintenance Cycle Stage 1 – Initiation:

a. Initiate the change request

Change request may be triggered for a variety of reasons i.e. enhancement need, software problem fixing, ad hoc request and from a wide variety of sources i.e. users, maintenance team and operator etc. All the request shall be initiated by completing a change request form which describe the detail of the requested change, reason for change together with requestor's information.

b. Conduct Initial Filtering

Upon receipt of the change request form from the requester, the head of ICT is responsible for conducting initial vetting and assessment on the change request. Incomplete/irrelevant change requests are filtered out and returned to the originator. Supported request are countersigned and implemented.

c. Review the Request

All change request forms will be reviewed by the head of ICT and maintenance team. A change request Ref. no. will be assigned and recorded on each Change Request for tracking purpose. Change requests will be considered in terms of their needs, urgency, benefits, etc. The type and priority of the change requests will also be assessed and revised if necessary. The accepted change request will be passed to next stage for impact analysis.

ii. System Maintenance Cycle stage 2 – Impact Analysis

a. Conduct Impact Analysis

Maintenance Team is responsible for thoroughly analyzing the accepted change requests on technical perspectives. The analysis shall assess the feasibility, scope, impact and potential ripple effects caused by the change as well as the possible solutions. Management Team shall also assure the impacts on business and user perspectives have been adequately analyzed.

b. Estimate The requires Resources

Based on the impact analysis result, resources required for the implementation of the requested change are estimated. Factors to be considered in the estimation include software and hardware, manpower, cost and schedule.

c. Document The Impact and required resources

The impact analysis result and resources estimation shall be documented in details in change request form for approval.

iii. System Maintenance Cycle Stage 3 – Disposition

a. Evaluate the Change Request

Based on the results from Stage 2, the Management Team evaluates the justification of the change request. Factors to be considered depend on the specific application system but shall generally include the following: tangible and intangible benefits gained from the change; total cost of the change; manpower required to implement the change; elapsed time needed to implement the change; disruption to current service; and retraining efforts for operation personnel and users.

b. Determine the Disposition

The management team shall finally reach a decision on the disposition of change request. The following are possible disposition;

- **Reject the change Request**

  If the change request is determined to be rejected, part of the change request form will be updated with the reasons for rejection and returned to the requestor.

- **Defer the change Request**

  If the change request is determined to be deferred for later re-consideration, change request will be updated regarding this. Deferred request will be brought by the management team for review when required.

- **Approve the Change Request**

  If the change request is determined to be approved, part of the change request form will be updated accordingly with the reasons for approval. Appropriate level of authority for approval shall be followed.

The management team will assess the approved request with the existing outstanding requests and assign an implementation priority or it. Final, the change request will be passed to the maintenance team for implementation.

iv. System Maintenance Cycle Stage 4 – Implementation

a. Design the Modification to the System

Based on the implementation priority assigned by the management team, the maintenance team will conduct detailed design for the change request. The impact analysis result and resources estimation produced in stage 2 will be used as input for the design work. In general, the main tasks include; identify the affected software modules, update the revised design of the system and develop implementation plan and test plan.

The implementation plan shall define the implementation and delivery arrangements for the proposed changes. In general, the plan shall cover the following area;

- Implementation and delivery approach;
- Implementation and delivery schedule;
- Implementation and delivery procedure; and
- Backup and recovery procedures.

The test plan shall state the testing requirements and arrangement for the proposed changes. In general, the plan shall cover the following areas;

- Testing approaches like unit test, integration test, user acceptance test, regression test etc.;
- Test resources like hardware, software, staffing;
- Test data and expected results;
- Testing, problem reporting, error correction and retesting procedures;
- Testing acceptance criteria; and
- Rollback plan should the change fail.

b. Build the Change into the System

The maintenance team is responsible for building the approved change into the application systems according to the design. After the modification are coded and unit-tested or at appropriate interval during coding, the modified software shall be integrated with the system and integration test shall be refined and performed. All effect of the modification on the existing shall be assessed and any unacceptable impacts shall also be noted. A return to the coding and testing may be requires to remedy such impacts.

The maintenance leader shall carefully monitor the progress of the implementation as well as the use of allocated resources. Any large discrepancy in the schedule and resource utilization shall be reported to the management team for consideration.

c. Test the Implemented Change

Once change has been made to the software modules, it shall be thoroughly tested to ensure the change is correct and does not introduce other errors to existing functions.

Once the change is completely tested on the technical aspect, it shall be tested on the user aspect where appropriate. The user acceptance tests shall be conducted by the user representative of the modified system this is to ensure that the implemented change is satisfactory to the customer.

d. Deliver the modified system

After successful testing, the implemented change could be delivered. To reduce the risk associated with the delivery of the change, the maintenance leader shall plan for and document the delivery procedures to ensure minimal impact on the user and the system due to unforeseen software failures not detected during testing.

When a change is a significant modification of user interface or functionality, user training shall be arranged.

The main tasks in the delivery processes involve conduction of physical check, notification of users, preparation of archives of the old system for backup and testing of recovery if required and conduction of installation and training.

2.2. THE GUIDELINES

Pursuant to the provisions of e-Government Act No.10 of 2019 and its Regulations, which direct on how and what to do with regards to efficient and effective e-Government implementation in the public sector, the Authority has prepared these guidelines to be used by public institutions. This document therefore, stipulates guidelines for software development, acquisition, operation and maintenance support.

2.2.1. General Guidelines

A public institution intending to undertake software development, acquisition, operation and maintenance shall ensure that:

i. Has competent internal ICT team with appropriate knowledge and skills.
ii. The ICT department/unit is involved in all related activities.
iii. ICT Staff are involved in all ICT related contract negotiations.
iv. The user department is involved in all related activities.
v. Application integration requirements are preferably based on open standards as guided by e-Government Application Architecture (e-GA/EXT/APA/001).
vi. Risks are appropriately managed.
vii. Proper criteria are used for estimating costs relate to license fees for all common use ICT support applications, such as antivirus, operating systems, office suites, systems/network monitoring and are appropriately planned and budgeted.
viii. Project change is done in accordance with institutional ICT project change management procedures.
ix. License scheme meets the user requirements where applicable.
x. There are appropriate change management processes.
xi. The internal ICT team is involved in the preparation and management of the respective contracts.

2.2.2. Software Development and Acquisition

2.2.2.1. General Considerations

A public institution embarking on application development/acquisition shall ensure that:

i. The applications fulfill institutional business requirements.

ii. It can only acquire an application from vendor when its ICT team lacks the required capacity for in-house development and has failed to get assistance from other public institutions.

iii. Shared systems fulfil stakeholders’ requirements.

iv. The applications shall not be platform dependent, such that development framework, database and operating system belong to one platform.

v. Knowledge transfer and training plans are part of the requirements.

vi. Requirements are prepared from re-engineered business processes.

vii. Security requirements, such as ability to generate and store audit logs, strong authentication and authorization, user management, session management, backup and recovery management are part of the application requirements.

viii. All system requirements are appropriately documented in a System Requirements Specification (SRS) document and verified by the user department.

ix. e-GA is consulted in case it is a common business support application such as human resource, finance/accounting, procurement, fleet management, payment gateways, e-office systems and e-mail systems.

x. It considers the use of open source technology.

xi. There are ‘test cases’ prepared with regards to the system requirement specifications and tested.

xii. The system design shall address the requirements of all user groups to promote digital inclusion, including persons with disabilities and users with limited access to digital services.

2.2.2.2. Guidelines for Software Development

A public institution intending to undertake in-house application development shall ensure that:

i. Security requirements have been considered during design stage and properly tested, including security vulnerabilities check prior to connecting to its network.

ii. Internal ICT team leads and maintains accountability throughout the development life-cycle, even where assistance has been sought from other public institutions.

iii. It develops Institutional application development standards, which must be verified by e-GA, in case of using other methodologies such as agile, extreme programming or rapid prototyping.

iv. It plans for data conversion and migration from the early stages, including defining migration scope and conducting full testing to ensure data accuracy and integrity whenever the application to be developed replaces the existing one.

v. It separates production, development and test environments, so as to ensure operational efficiency and effectiveness, including security.

vi. User requirements for the application are identified by the designated owner/custodian of processed information.

vii. There are mechanisms for tracking errors/bugs.

viii. A large/complex application is developed in phases.

ix. It uses the latest and stable technology supportable in the market.

x. It has appropriate license for tools that require license, and never allow the use of pirated ones.

xi. Test is performed in appropriate environment, properly documented and signed-off by the user department.

xii. User manual is prepared as part of the application documentation.

2.2.2.3. Guideline for Software Acquisition

a. Guidelines for acquisition of 'Off-the-Shelf' applications

A public institution intending to acquire Off-the-Shelf application to support their business operations shall ensure that:

i. It has properly customized it, either by in-house or outsourced experts, in order to fit in institution's operations.

ii. It has in place and operationalizes "project, vendor and contract management" practices.

b. Guidelines for acquisition of 'Software as a Service'

A public institution intending to acquire a software as a service (SaaS) and the related cloud software to support their business operations shall ensure that:

i. It intensively analyzes purchase contract and subscriptions requirements before acquiring such service.

ii. All Software as Service (SaaS) applications to be used by public institutions must be approved by e-GA prior to use.

c. Guidelines for acquisition of Outsourced Development

A public institution intending to outsource application development to support their business operations shall ensure that:

i. The consultant undertaking user requirements gathering shall not be the vendor that is engaged to develop the same system.

ii. A consultant who has participated in identifying and engaging an implementation vendor shall not participate in developing the application.

iii. A vendor does not bring any license cost for the outsourced development, except for maintenance and third-party software if applicable.

iv. A vendor does not access the production environment after completion of outsourced development unless explicitly authorized by the organization through a formal, time-bound approval process. Any approved access shall be strictly controlled, monitored, and limited to essential activities, with all actions logged and subject to audit.

2.2.2.4. Guidelines for Source Code Management

a. General Guidelines

A public institution shall ensure that:

i. Any contract related to system acquisition or software development must explicitly state that all source code shall be owned by the Government through the respective public institution.

ii. Source code delivered as part of system acquisition or outsourced development must be provided in open or non-compiled formats to allow full access to review, modify and maintain the software.

iii. All source code must be maintained in a secure, centralized version control system that supports full change tracking, branching, merging, and rollback capabilities.

b. Guidelines for Source Code Documentation

A public institution shall ensure that:

i. Source code developed or acquired includes clear, thorough, and well-structured documentation to facilitate understanding, maintenance, and future modification.

ii. Naming conventions, coding standards, and formatting guidelines used in the code are explicitly defined and documented for consistency and readability.

iii. Each file and script in the source code include header comments detailing its purpose, creation date, and any revision history.

iv. Any third-party libraries, frameworks, or tools used in the software are listed, including their versions, licenses, and purposes.

v. Documentation includes security considerations, such as secure coding practices, access controls, and any known vulnerabilities or mitigations.

vi. Documentation is regularly updated to reflect changes in the source code, ensuring it remains relevant and accurate.

vii. Source code changes are documented to ensure clarity, traceability, and ease of maintenance.

viii. Descriptive commit messages that accurately describe the purpose of each change for clarity and traceability are used.

ix. Repository structures, naming conventions and processes for accessing and maintaining the source code are documented.

x. README file in each repository is maintained to provide an overview of the project, dependencies and build instructions.

xi. Documentation of dependencies includes the purpose of each dependency, how it is managed, and any specific version constraints.

xii. Detailed change logs are maintained to document updates, fixes, and improvements for accountability and tracking.

xiii. A rollback plan is defined for each release to ensure issues can be addressed without significant downtime.

c. Guidelines for Source Code Storage and Version Control

A public institution shall ensure that:

i. Source code is stored in the source code repository hosted within a Government approved hosting environment.

ii. Sensitive information such as credentials is prohibited from being stored in the source code repository.

iii. It separates source code versions based on development, testing, and production to limit exposure.

iv. Branching strategies are used in version control to support collaborative development and minimize conflicts.

d. Guidelines for Source Code Access Controls

A public institution shall ensure that:

i. Source code ownership is clearly defined to clarify who is responsible for managing and maintaining the code.

ii. Access to the source code is restricted to authorized users only, in adherence to the principle of role-based access control.

iii. Audit trail is established to monitor and record all activities related to source code for security purposes.

e. Guidelines for Source Code Repository Systems

A public institution shall ensure that:

i. Source code repository structure is modular to simplify dependency management and facilitate scalability.

ii. Branch protection rules are implemented to ensure code changes are reviewed and approved before merged into the main (production) branch.

iii. Source code repository system is capable of integrating with approved continuous integration and continuous deployment (CI/CD) tools for streamlining code testing, building and deployment.

iv. Static code analysis tools are integrated into the source code management (SCM) systems to identify security vulnerabilities, coding errors, and compliance violations during the software development.

v. Regular backup of source code repositories is performed to prevent loss due to unforeseen circumstances such as hardware failure.

vi. Source code management (SCM) systems have a secondary site to ensure continuity of software development activities in the event of a disaster or system failure.

f. Guidelines for Source Code Maintenance

A public institution shall ensure that:

i. Periodic code reviews are conducted to improve quality, ensure adherence to standards, and identify potential security vulnerabilities.

ii. Dependency libraries are updated regularly to mitigate risks arising from vulnerabilities in third-party code.

2.2.2.5. Guidelines for Software Operation and Maintenance Support

a. General Consideration

A public institution embarking on application operation and maintenance support shall ensure that:

i. It uses ITIL for ICT and support services as guided by e-Government Infrastructure Architecture Standards and Technical Guideline (eGA/EXT/IRA/001), including presence of ICT Service Support Desk.

ii. The acquired or developed applications are hosted on its own approved equipment room or on Government data centers.

b. Guidelines for Application Operation Support

A public institution embarking on application operation support shall ensure that:

i. It is done using Government's internal capacity, or obtain e-GA approval in case of outsourcing the service from the vendor.

ii. There is a separation of duties between application developer and application administrators who perform day to day operations of the application such as preparing servers, installing and configuring software, loading data, restating failed instances, accommodating changes required by users, organizing maintenance, and minimizing downtime.

iii. All business operations in applications such as adding or removing users, reviewing users' roles and activities, and viewing or approving business transactions are done by business users and not ICT staff.

iv. All applications are supposed to be operated by the ICT department and be assigned application administrator(s).

v. There is Operation Level Agreement (OLA) between ICT and user departments stipulating key responsibilities of each.

vi. It handles its ICT security operations, such as backup and restoration, logs management and vulnerability management.

vii. ICT security operations such as vulnerability assessments and penetration testing are mandatory and done by the institution itself or a competent public institution approved by e-GA.

viii. The application undergoes security assessment at least once annually and have in place a documented process for all application operations, such as user and access management.

ix. Applications include copyright protection to safeguard intellectual property rights and to ensure proper use, reproduction, and distribution of software, source code, documentation, and digital content.

c. Guidelines for Applications Maintenance Support

A public institution embarking on application maintenance support shall ensure that:

i. Any changes, such as bugs and errors fixing, patches and upgrades are undertaken in accordance with change management process and are properly documented.

ii. Has in place documented procedures to guide any changes to application codes.

iii. Application developers do not interact directly with the production environment when implementing code changes, feature updates, or configuration modifications.

iv. User satisfaction mechanisms with regards to performance of application services are instituted.

v. All applications, which require licenses, have a valid maintenance license for patches and upgrades, including presence of a clear Service Level Agreements (SLA).

vi. Application changes and versions control are appropriately managed, including undertaking of security tests before incorporation of the changes into the live environment.

vii. Available upgrades and patches are regularly checked.

3. IMPLEMENTATION, ENFORCEMENT AND REVIEW

This document shall be:

3.1. Effective upon being reviewed and approved by the Authority Board of Directors.

3.2. Subjected to review at least once every three years or whenever necessary changes are needed.

3.3. Continually complied to and any exception to its application must be duly authorized.

4. RELATED DOCUMENTS

4.1. e-Government Guidelines ***(PO-PSM, 2017)***.

4.2. e-Government Application Architecture – Standards and Technical Guidelines ***(eGA/EXT/APA/001)***.

4.3. Government Software Applications Quality Assurance Guidelines and Checklist (**eGA/EXT/APA/002**)

4.4. Quality Assurance Compliance Guidelines for e-Government Applications ***(eGA/EXT/APA/007)***.

5. DOCUMENT CONTROL

Version	Name	Comment	Date
Ver. 1.0	e-GA	Creation of Document	November 2018
Ver. 1.1	e-GA	Inclusion of platform dependence on section 2.1.1.5 (ii) and on section 2.2.2.1 (iv)	November 2019
Ver. 2.0	e-GA	Aligning the document with e-Government Act of 2019	July 2022
Ver.3.0	e-GA	Reviewed and improved section 2.1.2.1 to 2.1.2.7, addition of section 2.2.2.4: Guidelines for Source Code Management and merging of the Standards for Development, Acquisition, Operation and Maintenance of e-Government Applications and Guidelines for Development, Acquisition, Operation and Maintenance of e-Government Applications into a single document	February 2026