THE REVOLUTIONARY GOVERNMENT OF ZANZIBAR

PRESIDENT'S OFFICE, CONSTITUTION LEGAL AFFAIRS PUBLIC SERVICE AND GOOD GOVERNANCE

e-GOVERNMENT AGENCY

Document Title

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

Document Number

eGAZ/EXT/APA/005

APPROVAL	Name	Job Title/ Role	Signature	Date
Approved by	Said Seif Said	Managing Director		DEC 2022

Table of Contents

1. OVERVIEW 1
1.1. Introduction 1
1.2. Objectives 1
1.3. Rationale 1
1.4. Scope 2
2. THE STANDARDS 2
2.1. Application Development Standards 2
2.1.1. General Standards 2
2.1.2. Requirement Gathering Standards 2
2.1.3. Requirements Analysis Standards 3
2.1.4. Design Standards 5
2.1.5. Development Standards 7
2.1.6. Testing Standards 9
2.1.7. Implementation Standards 11
2.1.8. Post Implementation Standards 12
2.2. Application Acquisition Standards 12
2.2.1. Identifying, Planning and Justifying the application Requirements 13
2.2.2. Restructuring Information System Architecture 13
2.2.3. Identifying a Development Alternative 14
2.2.4. Conducting a Feasibility Analysis 14
2.2.5. Performing the Selection Procedure 15
2.2.6. Proposal Evaluation Procedure 16
2.2.7. Implementing the Selected Solution 18
2.2.8. Key factors in Selecting Available Alternative 18
2.3. Application Operations and Maintenance Standards 19
2.3.1. System Maintenance Cycle 19
2.3.2. System Maintenance 19
3. IMPLEMENTATION, ENFORCEMENT AND REVIEW **24**
4. GLOSSARY AND ACRONOMY **24**
5. RELATED DOCUMENTS **25**
6. DOCUMENT CONTROL **25**

1. OVERVIEW

1.1. Introduction

Zanzibar e-Government Agency (eGAZ) is a public institution establish by the Zanzibar e-Government Agency Act No. 12 of 2019. The Agency is mandated to Coordinate Oversee and Promote e-Government initiatives and enforce e-Government related Policies, Laws, Regulations, Standards and Guidelines in Public Institutions. In executing its duties, eGAZ shall implement and maintain coordinated government operations for Information and Communication Technology (ICT) that include the formulation of standards, technical guidelines and procedures to effectuate the purposes of the Agency.

This document establishes standards to be adhered by public institutions during application development, acquisition maintenance and operations.

1.2. Objectives

The objective of this document is to describe the standards; which apply when developing, acquiring, operate and maintain applications in Public Institutions, focuses on and outline specific standards that must be followed when building applications. The application development, acquisition, operation and maintenance standards document is directed at application developers who will be designing, developing, and maintaining applications for their institutions. This includes external contractors, consultants, and business partners.

1.3. Rationale

This document interprets current industry standards and recommends an application development standard for adoption in the Public Institutions for the software/application development lifecycle. Application development standard will provide:

i. Adequate application development standards for all stages of application/software development process;
ii. Minimum requirement for application/software development activities, deliverables and acceptance sign-off; and
iii. A general measure for ensuring the application/software development methodology is in compliance with the application/software development standard.

1.4. Scope

The application/software development standards will highlight key characteristics of a software development lifecycle methodology and provide guidance for generic a waterfall development and iterative development.

This application development standard, acquisition, operation and maintenance will be used by all application/software developers across the Government i.e. Ministries, Departments and Agencies as well as Local Government Authorities.

2. THE STANDARDS

2.1. Application Development Standards

2.1.1.General Standards

2.1.1.1. Issue a written statement establishing and/or selecting agile or iterative 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.

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

2.1.2. Requirement Gathering Standards

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

i. Review existing systems/process;
ii. Describe data/system/process;
iii. Identify problem, areas/opportunities;
iv. Identify user needs/wants;
v. Conduct interviews;
vi. Identify manual and automated processes;
vii. Draw conceptual flow;
viii. Identify follow on projects/phases;
ix. Identify inputs (functional description);
x. Data entry screens; and
xi. Inputs from outside sources.

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

i. Document requirements;
ii. Document assumptions;
iii. Document outstanding issues;
iv. Estimate data storage requirements;
v. Identify legislative/contractual/security/privacy/access requirement;
vi. Document reporting requirements;
vii. Establish training requirements;
viii. Conduct initial walkthrough; and
ix. Obtain sign-off and approval.

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

i. For Waterfall Methodology business requirement Document is required; and
ii. For iterative methodology use enterprise architecture artifacts to further refine use-cases, models, and scenarios.

2.1.3. Requirements Analysis Standards

2.1.3.1. Analyze Business and system requirement

2.1.3.2. The specific intended use of the system to be developed must be analyzed to specify system requirements. The system requirements specification should describe

i. Functions and capabilities of the system;
ii. Business, organizational and user requirements;
iii. Safety, security, information, privacy, interface, operations, and maintenance requirements; and
iv. Design constraints and qualification requirements. The system requirement specification must be documented.

2.1.3.3. The system requirements must be evaluated considering the criteria listed below. The results of evaluations must be documented.

i. Traceability to system requirements and system design;
ii. External consistency with system requirements;
iii. Internal consistency;
iv. Testability;

v. Feasibility of application design; and

vi. Feasibility of operation and maintenance.

2.1.3.4. Establish and document application requirements, including the quality characteristics specifications, described below;

i. Functional and capability specifications, including performance, physical characteristic, and environmental conditions under which the application is to perform;

ii. Interface external to the application module/component/service;

iii. Testing requirement;

iv. Privacy and security specifications, including those related to compromise of sensitive information;

v. Data definition and database requirements;

vi. Installation and acceptance requirement of the delivered application product of the operation and maintenance site(s);

vii. User documentation;

viii. User operation and execution requirements; and

ix. User maintenance requirements.

2.1.3.5. Provide the following minimum set of documentation as part of requirement analysis standards;

i. For agile methodology

a. Project charter

b. Project Business Case

c. RACI matrix

d. Work Breakdown Structure (WBS)

e. Risk and issue logs

f. Project communication plan

ii. For iterative methodology

a. Software requirement specification;

b. Analysis class;

c. Use-case model;

d. Use-case package; and

e. User-interface prototype.

2.1.3.6. 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.

2.1.4. Design Standards

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

i. Design system flow;
ii. Develop data model, Create physical data model;
iii. 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;
iv. Conduct design walkthrough;
v. Conceptual flow/procedures, Screen design; and
vi. Process Implementation.

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

2.1.4.3. Evaluate system architecture and requirements for the application/software. The results of the evaluations must be documented.

i. Traceability to the system requirements;
ii. Consistency with the system requirements;
iii. Appropriateness of design standards and methods used;
iv. Feasibility of the application module/component/services fulfilling requirements; and
v. Feasibility of operation and maintenance.

2.1.4.4. 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.

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

2.1.4.6. Develop and document a top level design for the database.

2.1.4.7. Develop and document preliminary versions of user documentation.

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

2.1.4.9. Evaluate the architecture of the application and the interface and database designs considering the criteria listed below. The results of the evaluations must be documented.

i. Traceability to the requirements of the application;
ii. External consistency with requirement of the application;
iii. Internal consistency between the application components;
iv. Appropriateness of design methods and standards used;
v. Feasibility of detailed design; and
vi. Feasibility of operation and maintenance.

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

2.1.4.11. Update the user documentation as necessary.

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

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

2.1.4.14. Evaluate the application detailed design and test requirements considering the criteria listed below. The results of the evaluations must be documented.

i. Traceability to the requirements of the application;
ii. External consistency with architecture design;
iii. Internal consistency between application components and application modules/units;
iv. Appropriateness of design methods and standards used;

v. Feasibility of testing; and
vi. Feasibility of operation and maintenance.

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

i. For waterfall methodology
   a. Architecture Design;
   b. System/subsystem Design;
   c. Application Architecture and design;
   d. Interface design;
   e. Database design; and
   f. Screen/report design.

ii. For Iterative methodology
   a. Design class;
   b. Design model;
   c. Design package;
   d. Software architecture document; and
   e. Use-case realization.

2.1.5. Development Standards

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

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

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

i. 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;
ii. Develop test cases, Unit test programs, Develop secure code; and
iii. Develop application/software documentation, Users guide, Turnover documentation, Training materials, Conduct initial turnover walkthrough, Schedule turnover dates.

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

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

2.1.5.6. Evaluate application code and test results considering the criteria listed below. The results of the evaluations must be documented.

i. Traceability to the requirements and design of the application;
ii. External consistency with the requirements and design of the application;
iii. Internal consistency between unit requirements;
iv. Test coverage of units;
v. Appropriateness of coding methods and standards used;
vi. Feasibility of application integration and testing; and
vii. Feasibility of operation and maintenance.

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

2.1.5.8. Integrate the application units and application components and test as the aggregates are developed in accordance with the integration plan. It should 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.

2.1.5.9. 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.

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

i. Traceability to the system requirements;
ii. External consistency with the system requirements;
iii. Internal consistency;
iv. Test coverage of the requirements of the application;
v. Appropriateness of test standards and methods used;
vi. Conformance to expected results;
vii. Feasibility of application testing; and

viii. Feasibility of operation and maintenance.

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

i. For waterfall methodology

a. Application Code;
b. Application documentation; and
c. User/operational manual.

ii. For iterative methodology

a. Software Development Plan;
b. Build Document; and
c. Implementation model.

2.1.5.12. 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.

2.1.6. Testing Standards

2.1.6.1. 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:

i. Application test, software test, run parallel test and document results;
ii. User acceptance test, security test, develop test procedures and document results; and
iii. Issue production readiness recommendation.

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

i. Test coverage of the requirements of the application;
ii. Conformance to expected results;
iii. Feasibility of system integration and testing, if conducted; and
iv. Feasibility of operation and maintenance.

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

2.1.6.4. Upon successful completion of the test:

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

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

2.1.6.5. 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 should 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.

2.1.6.6. Provide the following minimum set of documentation as part of testing standards;

i. For waterfall methodology

a. Application test plan;
b. Testing scripts/scenarios;
c. Application testing procedures;
d. Software test reports;
e. Defect/error reports; and
f. Documented changes to the system.

ii. For iterative methodology

a. Defect;
b. Test case;
c. Test evaluation report;
d. Test plan;
e. Test result; and
f. Documented changes to the system.

2.1.6.7. 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.

2.1.7. Implementation Standards

2.1.7.1. Develop a plan to install the application product in the target environment as designated. The resources and information necessary to install the application product should be determined and be available. The project team should 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.

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

2.1.7.3. Support acceptance review and testing of the application product. Acceptance review and testing should 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.

2.1.7.4. Complete and deliver the application product.

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

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

i. For waterfall methodology.

a. Application user manual;
b. Application operator manual; and
c. Implementation document.

ii. For iterative methodology.

a. Implementation model.

2.1.7.7. 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.

2.1.8. Post Implementation Standards

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

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

i. That application objectives are being achieved; and

ii. That the application is achieving user's needs.

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

2.2. 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 IT 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 should consider a value-risk matrix to determine which options can be applied. The application acquisition should involve the identification and analysis of alternative solutions that are each compared with the established public institution requirements. Normally e-GAZ is not involved in providing advice in part of application acquisition.

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

2.2.1. Identifying, Planning and Justifying the application Requirements

One of the most essential assessments in decision making is identifying the institution objectives after first knowing the problem being solved.

One of the biggest challenges public institution face in procuring the application is to define the application requirements. Application requirement describe the objectives of the application. They define the problem to be solved, institution and application goals, application process to be accomplished, user expectations, and the deliverables for the application. Also the requirement should incorporate information about application inputs, information being processed in the application and the information expected out of the application. Each of this information should be clearly defined so that later gaps in requirements and expectations are avoided.

Application requirements can be gathered through interviews, questionnaires, existing system derivation, and benchmarking with related application.

The output of this stage is a decision to go with specific application, timetable, budget, and application expectations.

2.2.2. Restructuring Information System Architecture

After a thorough application analysis approach, an institution which is still in the progress of acquiring an application should remodel its information system architecture. Information system architecture is the conceptualization of how the institution information objectives are met by capabilities of the specific applications.

This structural design however describes the flow of the information, data hierarchy, application functionality, and technical feasibility.

The output of this stage should be a strategic planning level on how to develop specific application that meets the constrained defined by information system architecture.

2.2.3. Identifying a Development Alternative

After developing an information system architecture design an institution must select the development method of the application they intend to acquire.

There are several options in procuring application solutions. Some available alternatives are developing the application in-house, off-the self-solutions (purchasing commercially available solution), buying a custom made system for a vendor, leasing software from an application service provider or leasing through utility computing (contracted development), and outsourcing a system from other institutions.

While the Public Institution is in the phase of deciding which alternative being selected, the management should carefully examine not only the advantages and disadvantages of each procuring option, but importantly, the option must be best-fit with institutions business plan that has been documented. Any application development project, whether the application is built in-house or purchased elsewhere, should support the institutions business and IT strategy.

2.2.4. Conducting a Feasibility Analysis

As part of assessment in acquiring the solutions, a feasibility analysis is important to identify the constraints for each alternative from both technical and business perspective. Feasibility analysis incorporates the following categories:

2.2.4.1. Economic Feasibility

Analysis provides cost-benefit justification with being regard to the expenses of an application, which include procurement, project-specific, start-up, and operation costs. Some cost examples are one-time and recurring cost, consultants, support staff, infrastructure, maintenance, training and application software cost, this feasibility ensures that the solution won't exceed the budget limit as well as it increase the efficiency and better resource utilization.

2.2.4.2. Technical Feasibility

Assessment analyzes the technical reasonableness of the proposed solution. Technical feasibility evaluates whether the institution has the infrastructure and resources including hardware, software, and network capability to support the application. It also assesses the consistency of the proposed application in terms of technical requirement with institution technical resources. This assessment guarantees the reliability and capacity for the future growth.

2.2.4.3. Operational Feasibility

An evaluation reviews the extent of institution changes required to accommodate the proposed application. The proposed application should solve the business problems and provide better opportunity for the business since the business process might be changed.

2.2.4.4. Legal and Contractual Feasibility

The proposed solution must pass any related legal or contractual obligations associated with Public institution acquiring the application and should ensure that there are no illegal practices corresponding to the new application related with Public Procurement Regulation Authority.

Upon completion of the feasibility analysis, the risk analysis review must be conducted. Risk analysis evaluate the security of proposed application, potential threats, vulnerabilities, impacts, as well as the feasibility of other controls can be used to minimize the identified threats.

2.2.5. Performing the Selection Procedure

Selection procedure is the process of identifying the best match between the available options and the identified requirements. In this process, the public institutions request for a proposal from perspective providers, evaluate the proposal, and selects the best available alternatives. Public institution may request the proposal from providers by using various methods like request for information (FRI), request for bid (RFB), and request for proposal (RFP).

An RFI is used to seek information from vendors for a specific intention. RFI should act as a tool for determining the alternatives or associated alternatives for meeting the institution's needs.

An RFB is designed to procure specific items or services and used where either multiple vendor are equally competent of meeting all of the technical and functional specifications or only one provide can meet them.

Meanwhile the RFP specifies the minimal acceptable requirements, including functional, technical, and contractual aspects. An RFP's can be a lead to a purchase or continued negotiation.

2.2.6. Proposal Evaluation Procedure

Proposal evaluations is a crucial process in the application acquisition since one of more key stakeholders' reviews submitted proposals using a list of objective selection criteria and decide the best match between the product features and functionality with the identified requirements.

During this activity a public institution should follow these six steps in selecting a software vendor with their application package:

2.2.6.1. Examining potential vendors background

Potential software provider can be identified from software catalogs, list provided by hardware vendors, technical and trade journals, or consultants experienced in the other organizations, institutions and companies and web searches. These preliminary evaluation criteria can be used to pre-eliminate the unqualified potential vendors based on the vendor track record, reputation, and some previous feedback.

2.2.6.2. Determine the evaluation criteria

One of the most difficult tasks in evaluating the vendor and a software package is to determine a set of detailed criteria for choosing the best vendor and package. These criterial can be identified from the RFP feedback sent by vendors. Some areas that should be considered are characteristics of the vendor, functional requirements of the system, technical requirements, total project costs, scalability of the solution, project time frame, quality of the documentation provided, and vendor support package.

2.2.6.3. Evaluating providers and their applications

The objective of this evaluation is to determine the gaps between the institution needs and the capabilities of the vendors and their application packages.

2.2.6.4. Selecting the provider and its solution

Choosing the vendor and its software depends on the nature of the application. Negotiation can begin with vendors to determine how their packages might be modified to remove any discrepancies with the institutions application needs. Feedbacks from the users who will work with the system and the IT staff who will support the system have to be considered. Below is the list of criteria for selecting a software application package to use in order to meet institutions needs/requirements:

- 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.

2.2.6.5. Negotiate a contract

Once the vendor and its package selected, the institutions can move to the contract negotiation, in which the institution can specify the price of the software and the type of the support to be provided by the vendor.

The contract must describe the detailed specifications, all the included services provided by the vendor, and other detail terms of the system.

2.2.6.6. Establishing a service level agreement (SLA)

SLA is a formal agreement regarding the distribution of work between an institution and its vendor. Such agreement is created according to a set of agreed-upon objective, quality tests, and some what if situations. Generally, SLA defines institution and vendor responsibilities, framework for designing support service, and institution privilege to have most of the control over their system.

2.2.7. Implementing the Selected Solution

Upon completion of the contract negotiation, an acceptance plan should be agreed by both the institution and the vendor so the new application can be ready to be installed or developed. No matter what options the institution choose, even when they decide to build their software in house, the institution will most likely have to deal with some vendor(s) and certain software that has to be purchased from some supplier(s).

2.2.8. Key factors in Selecting Available Alternative

2.2.8.1. Buying the applications (off-the-shelf solution)

Purchasing commercially available solutions requires that the institutions adapt to the functionality of the system. Buying an existing package can be a cost effective and time saving strategy compared with in-house development.

When selecting a vendor package, an institution should consider the following; vendor stability, system upgrades, customer support provided by vendor, hardware and software requirements, required customization of the software.

A buy option should be carefully considered to ensure all the critical features of the current and future needs are included in the package. Buying makes sense if an institution plan to keep something for a long time, but technology typically becomes outdates every two to three years.

2.2.8.2. Leasing the Applications

Lease option can result in substantial cost and time savings compared to buy option or in-house development. Leasing can be a good choice for the institutions that has very few ICT operations that cannot afford large investment in IT applications. Many common features that are needed by most institutions are usually incorporated in the leased package even though it may not always exactly include all the required features. Also regarding the shortage of ICT officers, institutions may choose to lease instead of develop software in-house. An institution that has many ICT operations may also prefer to choose this option since to evaluate the potential IT solutions before investing a heavy installment, especially in the long run.

2.3. Application Operations and Maintenance Standards

2.3.1. System Maintenance Cycle

System maintenance cycle of an application system commences when the system is accepted and put into operation. On-going monitoring and maintenance of the system have to be performed during the system maintenance cycle so as to ensure the performance and functionality of the system match with the requirement needs of the users.

2.3.2. System Maintenance

A number of maintenance activities which arise from problem report, enhancement request or ad hoc request may occur concurrently in the system maintenance cycle.

Maintenance cycle has four stages:

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

2.3.2.1. System Maintenance Cycle Stage 1 - Initiation:

i. 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 should be initiated by completing a change request form which describe the detail of the requested change, reason for change together with requestor's information.

ii. Conduct Initial Filtering

Upon receipt of the change request Form from the requester, The IT manager 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.

iii. Review the Request

All change request Forms will be reviewed by IT manager 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.

2.3.2.2. System Maintenance Cycle stage 2 – Impact Analysis

i. Conduct Impact Analysis

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

ii. 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.

iii. **Document The Impact and required resources**

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

2.3.2.3. System Maintenance Cycle Stage 3 – Disposition

i. **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 should 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.

ii. **Determine The Disposition**

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

a. **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.

b. **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.

c. 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 should 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.

2.3.2.4. System Maintenance Cycle Stage 4 – Implementation

i. 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 test plan and implementation plan.

Based on the analysis results, the Maintenance Team should identify the software modules to be modified. The system documentation should also be modified to reflect the new design. Test plan and implementation plan should be produced which clearly state how the testing, implementation and delivery of the changes are to be accomplished with a minimal impact to current users. The design will form the blueprint for subsequent modification of the system.

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

a. Testing approaches like unit test, integration test, user acceptance test, regression test etc.;
b. Test resources like hardware, software, staffing;
c. Test data and expected results;

d. Testing, problem reporting, error correction and retesting procedures; and
e. Testing acceptance criteria;
f. Rollback plan if the change fail.

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

a. Implementation and delivery approach;
b. Implementation and delivery schedule;
c. Implementation and delivery procedure; and
d. Backup and recovery procedures.

ii. 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 should be integrated with the system and integration test should be refined and performed. All effect of the modification on the existing should be assessed and any unacceptable impacts should also be noted. A return to the coding and testing may be requires to remedy such impacts.

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

iii. Test The Implemented Change

Once change has been made to the software modules, it should 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 should be tested on the user aspect where appropriate. The user acceptance tests should be conducted by the user representative of the modified system this is to ensure that the implemented change is satisfactory to the customer.

iv. 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 should 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 should 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.

3. IMPLEMENTATION, ENFORCEMENT AND REVIEW

3.1. This document shall be effective upon being signed on its first page.

3.2. This document shall be subjected to review at least once every three years or whenever necessary changes are needed.

3.3. In case of any exceptions to these standards, authorization should be obtained from same authority which authorized this document or the owner of this document.

4. GLOSSARY AND ACRONOMY

Term	***Description***
Application	Computer programs, procedures, rules and associated documentation and data pertaining to the operation of a computer system
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
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
Maintenance	To repair, change, or add to software product

5. RELATED DOCUMENTS

5.1. e-Government Guidelines ***(PO-CLAPSGG, 2022)***

5.2. Standards for Development, Acquisition, Operation and Maintenance of e-Government Applications ***(eGAZ/EXT/APA/005)***

6. DOCUMENT CONTROL

Version	**Name**	**Comment**	**Date**
Ver. 1.0	eGAZ	Creation of Document	December 2022