THE UNITED REPUBLIC OF TANZANIA
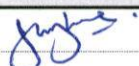
PRESIDENT'S OFFICE, PUBLIC SERVICE MANAGEMENT AND GOOD GOVERNANCE

e-GOVERNMENT AUTHORITY

Document Title

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

Document Number

eGA/EXT/APA/006

APPROVAL	Name	Job Title/ Role	Signature	Date
Approved by	Dr. Jabiri Kuwe Bakari	For: Director General		25/03/2020

Version: 2.0 – March 2020

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

Therefore, we call for all public institutions to effectively observe these guidelines when embarking on e-government initiatives relating to application development, acquisition, operation and maintenance.

Dr. Jabiri Kuwe Bakari
For: **DIRECTOR GENERAL**

Table of Contents

1. INTRODUCTION

1.1. Overview

e-Government Authority also known as "e-GA" is a public institution established under the e-Government Act No. 10 of 2019. The Authority is mandated to Co-ordinate, Oversee and Promote e-Government initiatives and enforce e-Government related Policies, Laws, Regulations, Standards and Guidelines in public institutions. The Act empowers e-GA to effectively formulate, manage and enforce public institutions compliance with e-Government standards and guidelines.

This document establishes guidelines for Development, Acquisition, Operation and Maintenance of Software used by public institutions in line with section 5(2) (c) and 22(3) of e-Government Act No.10 of 2019.

1.2. Purpose

The purpose of this document is to provide directives to all public institutions during application development, acquisition, operation, and maintenance activities in order to have quality and sustainable applications that facilitate improvement of internal business operations and service delivery.

1.3. Rationale

e-Government implementation, including application development, acquisition, operations and maintenance have been undertaken in uncontrolled approaches that resulted into duplications of efforts, silos initiatives, lack of systems integration, high cost of acquisition and operations and security concerns.

1.4. Scope

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

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 the general and specific guidelines for software development and acquisition as well as software operation and maintenance support.

2.1 General Guidelines

A public institution intending to undertake software development, acquisition, operation and maintenance should 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 Specific Guidelines

This part provides guidelines specific for application development, acquisition, operation and maintenance related activities implemented by public institutions.

2.2.1 Software Development and Acquisition

2.2.1.1 General Considerations

A public institution embarking on application development/acquisition should 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 should 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 using the approved template for System Requirements Specification (SRS) 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 owns source codes when it is specific for the institution.

xi. It considers the use of open source technology.

xii. There are 'test cases' prepared with regards to the system requirement specifications and tested.

2.2.1.2 Guidelines for Software Development

A public institution intending to undertake in-house application development should 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. Where necessary it may seek assistance from other capable public institutions and not from a vendor/freelancer.

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

iv. It adheres to the approved Government Applications Development, Acquisition, Operation and Maintenance Standards (e-GA/EXT/APA/005), in case application development methodology used is waterfall or iterative.

v. 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.

vi. It plans for data conversion/migration from the early stages, if the application to be developed replaces the existing one.

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

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

ix. There are mechanisms for tracking errors/bugs.

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

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

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

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

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

2.2.1.3 Guideline for Software Acquisition

2.2.1.3.1 Guidelines for acquisition of 'Off-the-Shelf' applications

A public institution intending to acquire Off-the-Shelf application to support their business operations should 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.

2.2.1.3.2 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 should ensure that:

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

ii. All SaaS are approved by e-GA.

2.2.1.3.3 Guidelines for acquisition of Outsourced Development

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

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

ii. The contract contains specific clauses that gives it source codes ownership and patent rights.

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

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

2.2.2 Guidelines for Software Operation and Maintenance Support

2.2.2.1 General Consideration

A public institution embarking on application operation and maintenance support should 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.

2.2.2.2 Guidelines for Application Operation Support

A public institution embarking on application operation support should 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.

2.2.2.3 Guidelines for Applications Maintenance Support

A public institution embarking on application maintenance support should 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. User satisfaction mechanisms with regards to performance of application services are instituted.

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

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

vi. Available upgrades and patches are regularly checked.

3. IMPLEMENTATION, ENFORCEMENT AND REVIEW

This document shall be:

3.1. Effective upon being signed by the Director General on its first page.

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

3.3. Consistently complied with, any exceptions to its application must duly be authorized by the Director General.

4. GLOSSARY AND ACRONYMS

4.1. Glossary

Term	Description
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.
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.

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.
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.

4.2. Acronyms

e-GA	e-Government Authority
ERP	Enterprises Resource Planning
ICT	Information and Communication Technology
OLA	Operation Level Agreement
SaaS	Software as a Service
SLA	Service Level Agreement
SRS	Systems Requirement Specification

5. RELATED DOCUMENTS

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

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

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

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

6. DOCUMENT CONTROL

Version	Name	Comment	Date
Ver. 1.0	eGA	Creation of the document	November 2018
Ver. 1.1	eGA	Inclusion of platform independence (2.1.1 section iv & 2.2.1 section ii)	August 2019
Ver. 2.0	e-GA	Aligning the document with e-Government Act No. 10 of 2019.	March 2020

APPENDIX: Approved Template for System Requirement Specification

This Template for System Requirements Specification (SRS) shall be used by all public institutions as guided in 2.2.1.1. (viii)

Software Requirements Specification (SRS) Template

Project Title	
Version Approved	
Prepared by	
Organization	
Date Created	

Revision History

Name	Date	Reason For Changes	Version

Table Contents

1.   Introduction

1.1   Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2   Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3   Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for

example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any

communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4.   System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1   System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behaviour defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

4.2   System Feature 2 (and so on)

5.   Other Non-functional Requirements

5.1   Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems.

Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>