UNIVERSITY OF DAR ES SALAAM

Software Systems Acquisition Guidelines

Directorate of Information and Communication Technologies

June 2024

Plan Approval

Document name	Software Acquisition Guidelines
Version	UDSM/ICT/SAG/V1.0_2024
Prepared by	Directorate of Information and Communication Technologies
Owned by	University of Dar es Salaam
Approved by	University Council
Date Approved	
Signature	
Signed by	

Document information

Document Name	Software Systems Acquisition Guideline
Category	Software Applications Development and Implementation
Related policies, standards, and guidelines.	• UDSM ICT Policy – Reference to section 3.2 *(Information Systems and Application Software)* • UDSM ICT Security Policy • UDSM Business Continuity and Disaster Recovery Plan • UDSM ICT Master Plan • UDSM ICT Maintenance Plan • Identity and Access Management Policy Records & Information Management Policy IT Risk Management Policy • Standard and Guidelines for Government ITCT Project Implementation. • ISO/IEC 25010 – Systems and Software Engineering- Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software quality Models • eGA/EXT/APA/005 - Standards for Development, Acquisition, Operation and Maintenance of eGovernment Applications (2022). • eGA/EXT/APA/007 – Quality Assurance Compliance Guidelines for e-Government Applications (2018)

Details of Version History and Authors

UDSM/ICT/SAG/V1.0_2024	*First Draft Document Submitted for Approval*	*DOICT Team*

TABLE OF CONTENTS

ACRONYMS AND ABRIVIATIONS

CCC&SC	Chief Cooperate Counsel and Secretary to Council
CD	Compact Disk
DOICT	Directorate of Information and Communication Technologies
DoICT	Director of Information and Communication Technologies
eGA	e-Government Authority
ICT	Information and Communication Technologies
ICTSC	Information and Communication Technologies Steering Committee
SAG	Software Acquisition Guidelines
SDD	System Design Document
SDLC	Software Development Life Cycle
SRS	Systems Requirements Specifications
UAT	User Acceptance Testing
UDSM	University of Dar es Salaam

1. INTRODUCTION

This document presents the Software Acquisition Guidelines (SAG) of the University of Dar es Salaam (UDSM). The guidelines are concerned with the acquisition and maintenance of software applications at UDSM. These software applications help the University to perform its core business and operations. For UDSM to acquire and maintain quality and secured software applications, a systematic and disciplined set of formal steps has to be observed. Hence, this guideline intends to provide guidance on important considerations and steps for any software applications acquisition at UDSM.

2. THE PURPOSE AND RATIONALE OF THIS GUIDELINE

The purpose of the UDSM software application acquisition and maintenance guideline is to ensure the University is implementing scalable, resilient, modular, quality, and maintainable software applications to support its core business and administrative functions. The guideline ensures that software applications are successfully designed, developed, implemented, supported, and maintained within time, cost, scope, and process limitations and a thorough consideration of associated risks, security and other software quality requirements. The guideline is supported by the following specific objectives:

i. To ensure proper software requirements specifications process are followed.
ii. To ensure business process reengineering is considered in all software acquisition projects.
iii. To ensure and facilitate implementation of software applications that consider software quality standard attributes.
iv. To ensure the secure and proper functioning of implemented software applications.
v. To ensure that software applications offer consistent and sustainable services through proper maintenance plans.
vi. To ensure software applications are supported and maintained as per changing business and operational requirements.

3. SOFTWARE APPLICATIONS ACQUISITION APPROACHES

3.1. Definition of key terms

- **Software project:** a project comprised of a group of people working together to specify, design, develop, test, and implement, deploy, or operationalise a new software application for a customer who might be either internal or external to the University.
- **Software enhancement project:** the type of project arises when the University wants to

enhance its existing software applications to provide new features or functions or perhaps to meet some external demands, like compliance with legal or regulatory requirements.

- **Software quality:** refers to the degree to which software applications satisfy the stated and implied needs of its various stakeholders, and thus provide value. Software quality is measured by quality properties with associated quality measures in two groups:
  - **product quality** attributes which define eight static properties or characteristics of software applications as indicated in the Appendices, and
  - **quality in use** attributes define five characteristics related to outcomes of interaction with the software product. Software quality attributes are provided in the Appendices.
- **Software Security: t**he degree to which a software application protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and level of authorisation.
- **Software user department:** any UDSM unit in which a core business function or business process requires the acquisition of a software solution.
- **Software Contractor:** an individual UDSM software developer, or an internal UDSM team formed to develop software, or a software firm contracted by UDSM to develop software.
- **Software ranking:** the categorisation of UDSM software solutions to be developed based on their scale or size and criticality *(e.g., small, medium, large, complex, risk levels.)*.

3.1. Software Acquisition Approaches

The software acquisition process may be implemented in several ways:

i. **In-house software development**: This approach involves the development of new applications or implementing major enhancements of existing applications, by an internal software development team, when major changes in supported business processes are introduced.
ii. **Outsourcing software development:** This approach involves contracting a software development firm or external software development expert(s) to develop new applications or make major enhancements to the existing applications in response to major changes in supported business processes or new requirements.
iii. **Package implementation:** This involves implementing pre-existing off-the-shelf or proprietary software packages.
iv. **Software customisation:** This involves the customisation of open-source or proprietary software packages to meet the specific software needs of the University.

4. IN-HOUSE AND OUTSOURCING SOFTWARE DEVELOPMENT APPROACHES

The development of software applications either by in-house development or out-sourcing development approaches should adhere to the following considerations and processes:

4.1. Generic considerations

i. Every software project to be implemented must have a complete project proposal that stipulates among other things project goals, project team, project manager, work plan, budget, schedule, deliverables, risk analysis, stakeholders, communications strategies, monitoring and control, and the main deliverable(s).

ii. The user department and all other interested parties of a proposed software to be acquired MUST be actively involved in the software development process.

iii. All software solutions to be developed should prioritise security considerations and implement relevant security measures from the initial stage of the development process to the end.

iv. All communication regarding the software development process must be in writing and every decision made must be documented.

v. Development of a new software solution MUST be accompanied by the analysis of available resources, technologies, and operating environment (e.g., hosting environment, server capacities, bandwidth, and security).

vi. Software quality attributes presented in the appendices should be considered throughout the Software Development Life Cycle (SDLC) for all software solutions.

vii. Any software application meant to be used by many users within and outside UDSM should be user-friendly and simple to use without the need for formal training. The need for training should only be for users with administrative roles or special tasks to perform.

viii. Integration and interoperability requirements should be considered as necessary for any software application to be used at UDSM.

ix. Large and complex software applications MUST be developed using modular designs.

x. A new software application is to be developed only if:

   a. there is a need of innovating a new solution, OR

   b. there is no alternative existing software, either at UDSM or in other public institutions that meets the requirements, OR

   c. the cost of purchasing an existing package from another institution is higher than the development cost, OR

   d. there is no open-source software that meets the requirements, OR

e. customising an existing similar software owned by other public institution(s) or open-source software is higher than the cost of developing the new one.

4.2. Software Acquisition Process

4.2.1. Solution ideation or request for acquiring software:

i. A user department has to write a concept note to DoICT explaining the need for acquiring a new software solution.

ii. The concept note should include the justification for the software, the value it would add to the respective unit and the University, and the expected source of funding.

iii. DoICT has to respond to the requesting unit by stating, with reasons, whether the request is accepted or rejected and advice on the way forward.

4.2.2. System authorisation activities:

i. The user department MUST seek authorisation from the appropriate Data Custodian (head of the relevant unit) and the ICT Steering Committee (ICTSC) or its chairman (after consulting DoICT).

ii. The user department under the guidance of DoICT, MUST develop a full software project proposal and submit it to the appropriate Data Custodian.

iii. DoICT shall advise the UDSM ICT steering committee or its chairperson about the relevant modality or approach to be used to acquire the software.

iv. Once the approach is decided, DoICT will inform the user department and how it would participate throughout the software acquisition process.

4.2.3. Appointment of a software Contractor

i. The user department and DoICT have to prepare high-level system requirements (user's needs) to be used to identify the software contractor.

ii. The ICTSC has to appoint the relevant software contractor as guided by the UDSM ICT acquisition and contract management guidelines and related procurement regulations.

iii. The software contractor MUST prove to have the relevant skills and expertise to develop the software.

iv. UDSM and the software contractor have to agree on the terms and conditions of the assignment and sign a contract prior to the start of any software acquisition activity.

v. A decision has to be made about source-code ownership and has to be included as part of the contract.

vi. DoICT should represent the University in drafting and managing any software acquisition contract to ensure the contractor delivers.

4.2.4. User requirements specification activities

i. The software contractor MUST collect and analyse system and user requirements before starting the software implementation process.

ii. All user departments and expected system users should be consulted in the requirements specification process.

iii. A system requirements specification (SRS) document MUST be produced before starting the development phase.

iv. As part of requirements specification and analysis, business process reengineering MUST be performed and documented.

v. The SRS MUST be validated and approved by the user department and DoICT before the start of the next phase.

4.2.5. Technical Design and Coding Activities for Development Approach

i. The contractor MUST produce an initial detailed system design document (SDD) before the start of the software coding process.

ii. The contractor MUST use and document an established software development methodology.

iii. A software development methodology to be used should be approved by DoICT based on the size, criticality and complexity of the software to be developed.

iv. The contractor MUST adhere to relevant and appropriate principles and methods based on the type of the chosen software development methodology.

v. Prepared system diagrams or architectures or flowcharts to support requirements and design specifications documents.

4.2.6. Change Management

i. Any change made to original requirements at any stage of software development must be documented by updating the original SRS.

ii. Any change introduced to the original system design must be documented by updating the SDD.

iii. An updated change management register should be kept throughout the SDLC for all changes made to the SRS document, SDD and source code.

4.2.7. System testing and acceptance testing procedures

i. System testing should be thoroughly done to check whether system components are compatible, interact correctly and transfer the right data to the right destination.

ii. Several tests MUST be performed including requirements, unit, release, scenario, performance test, and user acceptance test (UAT).

iii. For each test scenario to be performed, a testing case (protocol) and performance criteria should be prepared and a detailed testing report should be produced.

4.2.8. Quality Management, Monitoring and Control

i. The software development process should be monitored throughout the SDLC with a proper established schedule.

ii. DoICT should conduct progress meetings with the software contractor with a proper established schedule and deliverables.

iii. Software quality attributes in use and software product quality attributes should be assessed during software testing.

iv. All software development activities should be documented using appropriate documentation standards.

v. Ensure deliverables are submitted on time and the project is implemented within quality constraints of scope, time/schedule, and cost with the necessary attention to risks, product quality, and other resources needed for the project.

4.2.9. Piloting of a Software project

i. Once the software application has passed the UAT, pilot it in its business environment to test its performance and whether it meets users' needs.

ii. Prepare a proper piloting plan and schedule to make sure piloting takes not more than three months subject to the size of the software.

iii. Conduct user training either before deploying the software for use for large and complex systems or on-job training during software use for small and simple software.

iv. Access and availability of the software from different access points or locations and by using different connectivity, devices, and support application software to be used.

v. Collect feedback from the pilot users and immediately effect the proposed changes or correct observable errors.

vi. Use relevant criteria to solicit opinions from users who participated in the pilot about the stability and reliability of the software and whether it meets users' requirements.

vii. Prepare a deployment plan to operationalise the software solution.

4.2.10. Closure of Development Process and Hand-over

i. The software contractor MUST hand over the software product and readable source code to the University through DoICT and the user department.

ii. The software contractor MUST submit a full set of software documentation as part of the handover package. Such documents include:

- System Requirements Specification (SRS)document,
- System Design Document (SDD),
- Testing reports,
- Development report,
- Business process reengineering report,
- Piloting report,
- Training report,
- User manuals,
- Technical manuals,
- User training material
- Any agreed deliverables

iii. The contractor should return any UDSM facilities, devices, or tools given to him for development activities.

iv. UDSM shall retain full ownership of the system, including all hardware, software, and associated data. This ensures complete control over system operations, modifications, and future integrations.

v. A handover agreement should include a maintenance (liability) agreement for the period of six months for the contractor to provide support and address observed software defects.

vi. Final payment should be made to the contractor once all deliverables are delivered as agreed in the contract.

4.2.11. Deployment of the Newly Developed Software

i. Consider specific deployment activities, including the need for user training, based on the type of the software application to be deployed.

ii. Prepare relevant hosting environment to install the live system and deploy the system ready for use.

iii. Prepare a user helpdesk support to provide timely support to users when needed.

iv. Ensure network security measures are put in place to protect the system from network attacks and threats.

v. Ensure the software application is scanned for security vulnerabilities before its deployment.

vi. When applicable, register all users to the system or ensure self-registration is enabled as per the access level configurations.

vii. Make any other necessary preparations for users to start using the system to perform tasks.

viii. Inform relevant users of the system about the deployment and provide a specific date and time when the new system will be officially operational.

ix. If the software application is large, complex, and has many modules, start deployment by prioritising a few key modules and then scale up by incrementally and conveniently adding more modules.

x. Make sure the deployment is completed within schedule (within the liability period) and then prepare and submit the deployment report.

xi. The end of the deployment process marks the start of the maintenance phase. Implement maintenance procedures as provided in the UDSM ICT Resources Maintenance Guidelines.

4.2.12. Ownership of the system and intellectual property rights

i. UDSM shall hold all intellectual property rights to any customizations, modifications, or configurations developed as part of the system acquisition. This includes any unique processes, methodologies, or innovations created during the software project.

ii. The vendor must ensure that all components of the system comply with intellectual property laws and do not infringe on third-party rights. The vendor shall indemnify UDSM against any claims arising from IP infringements.

i. UDSM shall have perpetual, unrestricted licenses to use, modify, and distribute the system within the institution's operations.

5. IMPLEMENTATION OF PROPRIETARY SOFTWARE PACKAGES

5.1. Generic considerations for implementing off-the-shelf software packages

i. DoICT to prepare and keep an updated list of all proprietary software packages needed by different user departments.

ii. All acquisition of proprietary or off-the-shelf software MUST be requested through or by the DoICT.

iii. Proprietary software packages MUST be acquired in volume licencing arrangement

instead of single user/computer.

iv. If proprietary software has its equivalent as an open-source software, the latter should be prioritised and users encouraged to use it.

v. DoICT to prepare and keep an updated list of open-source software packages approved for use at UDSM.

vi. DoICT to prepare and keep an updated list of software suppliers approved by UDSM.

vii. Any purchased software package should meet the integration or interoperability requirements with existing software packages at UDSM (if applicable).

viii. An appropriate infrastructure environment should be prepared before the purchase of proprietary software.

5.2. The Implementation Process

5.2.1. Select the packages to implement

i. A user department to specify the application software to purchase based on requirements and submit a written request to DoICT.

ii. The user department to inform DoICT about the expected source of funds to be used for the purchase of the requested software.

iii. DoICT to establish the relevance of the needs and acquisition alternatives, if any.

iv. DoICT to recommend to the user department the appropriate software package to use if already exists at UDSM or as an approved open-source package, or to purchase if there are multiple options.

5.2.2. Vendor selection and contract signing

i. DoICT to prepare the software purchase order.

ii. DoICT to select the software vendor from the list of approved suppliers and recommend them for consideration to supply.

iii. DoICT to submit the list of recommended software packages and suppliers to the user department.

iv. The procedures for the acquisition of ICT resources at UDSM should be followed as per the UDSM ICT Resource Acquisition Guidelines and procurement regulations.

5.2.3. Purchase contract and package deployment

i. A software purchase contract is to be signed between UDSM and the software supplier.

ii. The contract should include, among other contractual terms, conditions for:

   a. payment arrangements,

b. deployment schedule and requirements,
c. liability period,
d. maintenance arrangement,
e. source code ownership (if applicable),
f. data migration (if applicable) and
g. the customisation requirements (if applicable).

iii. DoICT to prepare deployment environment, arrangement, and schedule.

iv. Follow the relevant procedure as provided in sections 3.2.9 and 3.2.11 of this Guideline.

5.2.4. Post-deployment agreement

i. Follow relevant procedures for the closure of the software implementation project as provided in sections 3.2.10 and 3.2.11 of this Guideline.
ii. Follow the procedure for the maintenance of software solutions as provided in the UDSM ICT Resource Maintenance Guidelines.

6. IMPLEMENTATION OF CUSTOMISED SOFTWARE PACKAGES

6.1. Generic considerations for customising software packages

i. The decision to customise an existing software is to be made only when:
   a. *doing so would produce more valuable software and add value,*
   b. *purchasing ready-made software would be more costly,*
   c. *development of new software would be more costly and take a long time to completion,*
   d. *the University or its units face emergency situations that prompt urgent needs for new software solutions,*
   e. *the university has relevant experts to customise the package and operational environment for hosting the software package.*
   f. *The software to be customised meets at least 70% of the requirements for a new system.*
   g. *Customisation decision should consider design complexity and technology compatibility.*
ii. A decision to customise an open-source software package or an existing software package owned by other public institutions or software firms must be approved by UDSM management after consulting DoICT.
iii. Make a thorough analysis of the software *(e.g., source code, design, development platforms, technologies, and documentation.)* to establish the feasibility of successful customisation before deciding to customise.
iv. The security, support, maintenance and sustainability of software should be ensured before making the decision to customise.

6.2. Customisation, piloting, and deployment.

Follow appropriate software acquisition procedures as provided in sections 3.2.1 to 3.2.11 of this guideline.

7. IMPLEMENTATION, REVIEWS, AND ENFORCEMENT

7.1. Implementation and Reviews

i. This Guideline shall come into operation once approved by the UDSM ICT Steering Committee and then shall be considered mandatory for all activities related to software acquisition and implementation at UDSM.

ii. Failure to observe this guideline may subject individuals to disciplinary action which may include termination of employment or contract, required to pay for the associated financial loss, or legal action or a combination of such actions.

iii. This document shall be reviewed after three years or at any time whenever the need for improving any part of these guidelines arises.

7.2. Exceptions

In case of any exceptions to this guideline, it shall be thoroughly documented and followed through a proper channel of authorisation using the same authority that approved this document.

7.3. Roles and Responsibilities

i. It is the responsibility of leaders of every unit at UDSM to read, understand, and adhere to these guidelines whenever they need a new software solution.

ii. Users are expected to exercise reasonable judgement in interpreting these guidelines and in making decisions regarding the acquisition of a new software solution.

iii. Any person with questions regarding the application or meaning of statements in this guideline shall seek clarification from the DoICT office.

iv. DOICT shall enforce compliance with this guideline in all matters related to the development or acquisition of a new software solution.

7.4. Monitoring and Evaluation

i. The UDSM ICT Steering Committee shall monitor and evaluate compliance with this guideline as part of its core responsibilities.

ii. DoICT shall conduct regular assessments to establish the need for improving this guideline or accommodating new requirements.

Appendices

Software Product Quality Characteristics & Sub-characteristics				Software Quality in Use Characteristic & Sub-characteristics	
Functional suitability		**Reliability**		**Effectiveness**	
	Functional completeness		Maturity	**Efficiency**	
	Functional correctness		Availability	**Satisfaction**	
	Functional appropriateness		Fault tolerance		Usefulness
Performance efficiency			Recoverability		Trust
	Time behaviour	**Security**			Pleasure
	Resource utilization		Confidentiality		Comfort
	Capacity		Integrity	**Freedom from risk**	
Compatibility			Non-repudiation		Economic risk mitigation
	Co-existence		Accountability		Health and safety risk mitigation
	Interoperability		Authenticity		Environmental risk mitigation
Usability		**Maintainability**		**Context coverage**	
	Appropriateness recognizability		Modularity		Context completeness
	Learnability		Reusability		Flexibility
Operability			Analysability		
	User error protection		Testability		
	User interface aesthetics		Modifiability		
	Accessibility		Testability		
Portability					
	Adaptability				
	Installability				
	Replaceability				