# FLYHUB (U) LIMITED
# SOFTWARE DEVELOPMENT FRAMEWORK

| Policy Details | |
|---|---|
| ***Policy Name:*** | Software Development Framework |
| ***Policy Type:*** | Governance Standard |
| ***Policy Level:*** | Country |
| ***Policy Owner:*** | Chief Executive FLYHUB |
| ***Policy Number:*** | HR/2021/1/V1, HC |
| ***Classification:*** | Internal Use Only |
| ***Effective Date:*** | 11/11/2021 |
| ***Amendment Date:*** | 03/11/2021 |
| ***Approved By:*** | FLYHUB Board |
| ***Approval Date:*** | 11/11/2021 |
| ***Next Review Date:*** | 11/11/2023 |

**Document Classification**

*Internal Use Only*

This document has been classified as for Internal Use Only and has been issued strictly for the internal business purposes of FLYHUB (U) Limited. Dissemination, by any means outside the organization, is prohibited without prior written approval sought from the Policy owner.

TABLE OF CONTENTS

## 1.0 INTROUCTION

Flyhub subscribes to sound corporate governance principles, one of which is the use and application of Policies and Governance Standards which define and articulate practices, boundaries and expectations within which the company will operate.

This framework enables the efficient development and deployment of quality and consistent software applications through the pragmatic establishment and execution of good software engineering practices and the adoption of innovative approaches and technologies.

## 2.0 PURPOSE OF THE FRAMEWORK

To define the methodology to be followed by Flyhub in developing and deploying software applications. This software framework will help in increasing team productivity and ensuring software application quality, robustness, and reliability.

## 3.0 APPLICABILITY OF THE FRAMEWORK

.

This software development framework will apply to all Flyhub software development projects, including those developed by Flyhub, developed with external partners and exclusively developed by partners.

## 4.0 SOFTWARE DEVELOPMENT PRINCIPLES

Flyhub will adopt the following software development principles in developing software applications;

i. Develop what matters to clients – carry out research when developing a software development proposal to ensure that the problem or need is important to a significant number of clients, can be solved using available technology and is therefore worth addressing.
ii. Client Consent - Ensure the client and Flyhub understand the scope of what we aim to build and the process on which we are embarking so that we have informed consent frm the client.
iii. Co-create software with clients – build prototypes and continuously test them with clients during the development process to ensure solution meets client requirements/needs.
iv. Adhere to software development best practice – as defined by International professional bodies and associations.
v. Simplicity – make software easy to use, maintain, upgrade and fix defects or failures
vi. Regulatory Compliance - Ensure compliance with prevailing regulatory requirements

## 5.0 SOFTWARE APPLICATION DEVELOPMENT PROPOSAL

Software development projects will be evaluated based on business cases which will include:

a. **Market and competitor analysis;** an assessment of market and competitive environment as relates to the proposed software application.

b. **Problem statement:** a description of the problem the software application will solve or the need it will meet.

c. **Proposed solution;** a detailed description of the proposed software solution including; how it is designed, how it will be developed, how it will work.

d. **Solution benefits;** a description of the benefits of the software application including; improving customer experience, increasing efficiency, cost reduction, cost optimization

e. **Strategic alignment:** a demonstration of how the software application is aligned to the Flyhub strategy.

f. **Financial impact;** a summary of the projected revenue and cost impact of the software application, including any assumptions made

g. **Risk assessment;** a review of the potential risks associated with the software solution and how they will be mitigated.

h. **Assumptions, Constraints and Key dependencies;** description of the factors, limiting conditions, circumstances or events that may impact delivery of the project and how they will be managed.

i. **Proposed timelines;** a draft roadmap indicating the various activities, milestones and timelines for delivery of the software application

j. **Approvals:** management approval indicated by appended signatures

## 6.0 SOFTWARE DEVELOPMENT IMPLEMENTATION

The following options will be available to Flyhub in the implementation of approved projects; leasing, buying, developing or partnering.

The following considerations will be considered in selecting the option to be adopted in implementing each software development project.

**a. Speed to market**

Approach selected will be one that can deliver the speed to market required by the client.

**b. Cost efficiency**

The approach selected will be one that delivers quality software at optimum cost.

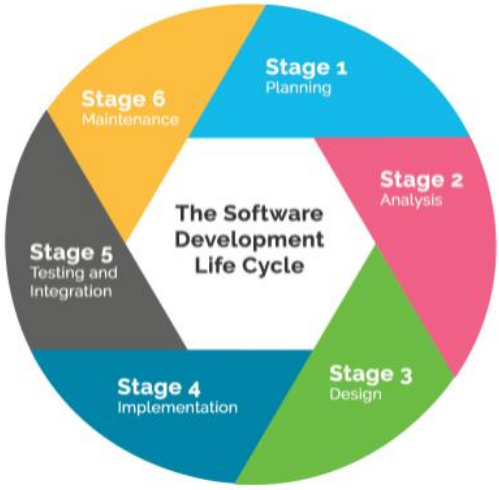
**c. Knowledge and Skill adequacy**

The approach selected will be one that brings to bear the appropriate level of knowledge and skill required to deliver quality software

## 7.0 SOFTWARE DEVELOPMENT

A software development methodology is a model for development of software applications. A software framework introduces a specific process to be followed in successfully designing, developing, testing, delivering, implementing, and maintaining software products.
Below are the stages typically followed in developing software:

### Software Development Life cycle




### Software Development stages

Any digital project consists of the planning, analysis, design, implementation, testing and integration, maintenance stages. A set of rules is applied to mentioned stages, upheld by all team players, is called software development methodology.

**Planning.** Defining the overall flow breaking down into small, easy manageable parts is the start. All elements are set in order to develop the project. An in-depth set of functionalities is identified for each element. Suitable technology is selected for the current product to be implemented while planning phase.

**Analysis.** It implicates identifying goals, demands eliciting, transforming facts, diagnosing issues related to the current project, recommending positive building. While analysis it is created requirements in order the scope of work to be defined.

**Design.** Stakeholders review documented specification for choosing a suitable approach. Applicable requirements are used for the product's architecture design with limited functionality; also, for identifying approaches to be applied based on the design document specification (DDS).

**Implementation** stage is a start point of project building. Code writing requires applying various programming tools, languages to match the documented specification. It helps in turning an idea into a future solution.

**Testing and Integration.** Bringing project's separate parts together into a test environment to be checked for bugs, other fails for meeting standard criterion is involved within mentioned stage.

**Maintenance.** Systematically testing functionalities for bugs that were not found or identified, for errors that are not shown in the fail logs, those are vital maintenance part that guarantees the project goes within the planned requirements.

The first step towards software development is to choose the best development methodology that suits your organization's culture, team size, flexibility status, and business and functional requirements.

While software development methodologies largely share the same stages, they have different approaches to navigate through these stages in order to address the shortcomings of their predecessors. The following are the benefits of adopting a software development methodology:

1. Increase efficiency by clearly defining the goals, objectives, and architectural roadmap.
2. Enhance efficiency, quality, reliability and robustness by adopting best practice
3. Reduce cost of software development by minimising resource wastage
4. Reduces effort required for software development as less programming and testing is required
5. Eases introduction of new design patterns and programming tools
6. Eases addition of new functionality
7. Shortens time-to-market by prioritizing high value tasks.

The following are the limitations of software development methodologies:

1. May not be adaptable enough to keep up with changes in the market
2. Mat not adequately address complexity
3. May introduce rigidity in the process

The following should be taken into consideration when selecting a software development methodology.

1. Client needs
2. Team's size
3. Scope and complexity of project
4. Probability that development requirements will change
5. Product risk profile
6. Time available
7. Budget available

The following are general guidelines on when to use each software development methodology:

| No | Project Characteristics | Recommended Software Development Methodology |
|---|---|---|
| 1 | Fixed requirements, time and budget are available | Waterfall Development |
| 2 | Requirements are variable and not definitive, incremental development is required | Agile Development |
| 3 | Large software project that focuses on features and have a good team strength to divide the feature-centric work | Feature Driven Development |
| 4 | Small project that needs to be delivered in a short time | Lean Development or Rapid Applications Development |
| 5 | Software requires continuous risk evaluation, and there is a reasonable time bracket for validating the release from users | Spiral Development |
| 6 | Development team is distributed and large and is working on an enterprise project. | Scaled Agile Framework |

**a. Project team**

i. After a project has been approved by Flyhub Management, the Chief Executive will assign team members to the project and will determine the order in which the projects will be developed depending on the expected benefits of the project.

ii. Each project team shall comprise of the following; Team Leader, Researcher, Software Developers, User Interface Designers/User Experience Designers, Software Quality Assurance and Software Testers

iii. The number of team members performing the above roles will depend on the scope and scale of the project and will be determined by Flyhub Management.
iv. Additional members may be added to the team depending on the specific requirements of the project.
v. The project teams will meet daily to review progress, assign tasks and highlight challenges in delivery to be addressed by team members or to be escalated to other parties outside the team.

### b. Software Design

The following activities will be completed as part of software design: defining overall architecture, software specifications, identifying hardware and network requirements.

### c. Software Coding

Software applications will be developed in the Cloud software development environment by Software Developers using approved Coding applications and tools as indicated below:

| DATABASE | Version | Type | Usage |
|---|---|---|---|
| PostgreSQL | | RDMS | Any |
| MySQL | | RDMS | Any |
| Microsoft SQL | | RDMS | Any |
| Oracle | | RDMS | Any |
| SQLite | | | Mobile Applications |
| MongoDB | | NoSQL | Any |
| Amazon RDS (Oracle, MSSQL, MySQL) | | RDMS | Cloud |
| Amazon Simple DB | | | Cloud |
| Amazon Dynamo DB | | NoSQL | Cloud |
| Microsoft Azure SQL Database | | | Cloud |

| Framework / Libraries | Version | Usage |
|---|---|---|
| React | | Web Applications |
| React-Native | | Mobile |
| Angular | | Any |
| Vue.JS | | Any |
| NodeJS | | |
| Xamarin | | Mobile Applications |
| .NET Core | | |
| Spring, Spring Boot | | |
| Django | | |
| Torch / PyTorch | | |
| Spark | | Machine learning framework |
| TensorFlow | | |
| Bootstrap | | |
| Others | | Laravel, Express, Ruby on Rails, |

| LANGUAGE | Version | Type | Usage |
|---|---|---|---|
| Java | | | |
| JavaScript | | | |
| TypeScript | | | |
| Kotlin | | | |
| SQL | | | |
| C# (.NET / .NET Core) | | | |
| Python | | | |
| PHP | | | |
| JSON, XML, HTML5, CSS | | | |
| C / C++ | | | |
| SWIFT / Object C | | | |
| Flutter / Dart | | | |
| Others | | | VB.NET, F# |

Software Developers will write code as per the software design specifications and will review the code on completion to verify that all software design specifications have been met.

All software code will be stored in a secure repository in the Cloud and not on individual developer work terminals/computers.

All software code will be saved and backed up to the Cloud every 2 hours to avoid loss of software code during software coding.

Software modules will be developed following the micro-services model whereby each module is developed as an independent unit. This approach will ease development, testing, maintenance, issue resolution (debugging) and upgrades without affecting other services.

### d. Quality Assurance

The code will be reviewed by a Quality Assurance Officer to check for compliance with design specifications, software quality standards and for any vulnerabilities. Any issues identified will be referred back to the Software Developer for resolution before deployment to the test environment for Testers.

Quality Assurance Officers will only have rights to view software code and will not be able to amend, copy or alter software code in any way.

### e. Software Testing

On passing Quality Assurance, software applications will be deployed to the Test environment and will be made available to Software Testers for testing.
Software Testers will develop user test cases based on the specifications defined in the business case and will approximate as closely as possible the live environment in which the software application will be used.
Software Testers will test the software applications against the Test cases developed, document test success and failure and share issues/bugs with Software Developers for rectification.
Software Testers will test software end to end once issues/bugs have been resolved or addressed by Software Developers.
Where specific test cases cannot be completed in the testing environment due to certain limitations, these cases shall be documented and marked for testing in the live environment before release of the software application.

On successful completion of testing, Software Testers will compile and submit test results to the Secretary of the Change Approval Board (CAB) for review and consideration by the CAB committee. The CAB committee will comprise of the Chief Executive (Chairman), Quality Assurance Officer, Senior Product Manager, Systems Architect, Administrator (Secretary) and the Project Team Leader.

Software Developers will develop and submit change deployment plans to the CAB Secretary, detailing how the software will be deployed to the live environment, any downtime that will be required and time required for the deployment.

**f. Software deployment**

On obtaining approval from the CAB committee and assignment of a change window, Software Developers will deploy software application to the live environment.

On completion of deployment to the live environment, Quality Assurance Officers will verify that the software application has been deployed as per design specifications.

Software Testers will carry out end to end tests in the live environment to confirm that the software application meets the user specifications.

Depending on the time required and complexity of the software application, software deployment may be carried out at once or staggered over a specific period of time as per the deployment plan.

## 8.0 SOFTWARE SECURITY

Flyhub will adopt practices below in order to reduce the number of vulnerabilities in released software, mitigate the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and address the root causes of vulnerabilities to prevent future recurrences.

a. Define criteria for software security checks
b. Protect all forms of code from unauthorized access and tampering by safeguarding the development, build, distribution, and update environments and following the least privilege principle
c. Provide a mechanism for verifying software release integrity by digitally signing the code throughout the software lifecycle
d. Design software to meet security requirements and mitigate security risks
e. Verify third-party software complies with security requirements
f. Configure the compilation and build processes to improve executable security
g. Review and/or analyse human-readable code to identify vulnerabilities and verify compliance with security requirements
h. Test executable code to identify vulnerabilities and verify compliance with security requirements
i. Configure the software to have secure settings by default
j. Archive and protect each software release
k. Identify, analyse, and remediate vulnerabilities on a continuous basis

## 9.0 SOFTWARE UPGRADES AND MAINTENANCE

Software applications will be reviewed as required and upgrades and maintenance will be developed and deployed following the same methodology as laid out for new software development.
Software application version management and tracking shall be carried out for all software applications.

## 10.0 SOFTWARE APPLICATION EVALUATION

The Software Application Development team will review the software developed and the effectiveness of the development process in order to identify areas of improvement to be adopted in subsequent software development projects.

## 11.0 SOFTWARE DECOMMISSIONING

Once software is determined to be obsolete and is replaced or is no longer required, the Software Applications Officer will generate a request to decommission the software application.

The software application decommissioning request will detail the software type, version, reason for decommissioning, alternative software application if any, communication plan to users and proposed date of decommissioning.

Software application decommissioning request along with a software decommissioning plan to the CAB committee for review and approval.

On obtaining approval from the CAB Committee to decommission software, Software Developers will archive the software application code and associated data to an archive.

The Software Applications Officer will also delete the software application and associated date from the live environment and confirm that the application is no longer active.

## 12.0 SOFTWARE DEVELOPMENT TRAINING

Flyhub Management shall be responsible for ensuring Software Developers receive training, coaching and certification on the software engineering methods, related processes and toolsets; and Once software is determined to be obsolete and is replaced or is no longer required, the Software Applications Officer will generate a request to decommission the software application

## 13.0 ADMINISTRATION

| | ***Details*** |
|---|---|
| *Title* | *Chief Executive* |
| *Name* | *Joel Muhumuza* |
| *Department* | *FlyHub* |
| *Telephone* | *+256 (0)312 224 600* |
| *Email* | *muhumuzaj@stanbic.com* |