



ፀሐይ ገንባ
TSEHAY BANK
ከሆኑ



Software Development Policy

(Draft Version 1.0)

Table of content

Table of content	i
CHAPTER ONE.....	1
1. Introduction	1
CHAPTER TWO	2
2. General context.....	2
2.1 Definitions	2
2.2 Policy authority	3
CHAPTER THREE.....	4
3. Scope, purpose, objective and statement of the policy	4
3.1 Scope	4
3.2 Purpose	4
3.3 Policy Objectives	4
3.4 Policy statement.....	5
3.4.1 Software Project Initiation	5
3.4.2 Software Project Team	5
3.4.3 Software Project Management.....	5
3.3.4 End User Team	6
3.3.5 Project proposal	6
3.3.6 Requirement Analysis.....	6
3.3.7 Time Analysis.....	6
3.3.8 System Design	7
3.3.9 Coding	7
3.3.10 Coding convention.....	8
3.3.11 Testing	8
3.3.12 Manual	8
3.3.13 Training	8
3.3.14 Deployment	8
3.3.15 Maintenance	9
CHAPTER FOUR	10
4. Applicability and use.....	10
4.1 Audience.....	10
4.2 Acceptable Use.....	10
4.3 Violation	10
CHAPTER FIVE.....	11
6.1 Sample Template for Requirement Analysis Document and Use Case.....	11

CHAPTER ONE

1. Introduction

Software is a general term primarily used for digitally stored data such as computer programs and other kinds of information read and written by computers to run business process. And Software development is the set of activities that results in software products. Software development may include research, new development, modification, reuse, re- engineering, maintenance, or any other activities that result in software products. Useful software systems are complex. To remain useful, they need to involve with the end users' need and the target client.

When we come to Tsehay Bank, as new bank there is need to develop new software and customization and other partners may require the integration of different software applications in order to facilitate their business process and access vital information to support their informational and collaboration activities. The integrity of the information as well as security and reliability must be assured via the strict application of methods and best practices of software development.

Currently IT has the potential for bringing about significant progress in the service delivery of the Bank, for which the software development wing has a dynamic and high contribution for the bank, due to this coordinated software development policy is need to meet the expectations of the Bank. And standardizing the software development process with different perspective like reduce the time of development, it will reduce the time spent correcting problems in the product line, increase software quality, it will give developers work with similar development standard and to manage software development process.

The intent of this document is to create a guideline for developing standardized and quality software. This document is not a description of a complete software development process or it is not text for software development. This document is simply a set of rules to follow during the development process that will help to produce a standardized higher quality result. The document is organized into five chapters. Chapter 2 is general context about the policy and defines terms used in this policy document. Chapter 3 specifies the

purpose and scope of this policy document. Chapter 4 articulates the policy statement and set the underlying principles and rules to be followed as developing software. Chapter 5 states procedures to be followed during different stages of software development.

CHAPTER TWO

2. General context

2.1 Definitions

In this document, the following words have the respective definitions,

Application Software: is computer software designed to help the user to perform singular or multiple related specific tasks.

Software Project Inception: is part of software development lifecycle in which you will decide to embark the development project.

Requirement: a requirement is a feature that the system must have or a constraint that it must satisfy to be accepted by the client or expected output from the system after development is finalized.

Functional requirements: it may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

Nonfunctional requirements: describes system-level requirements directly related to how system is operating to when execute task. This includes performance, security, modifiability, error handling, hardware platform, and physical environment. Scenario: a that are not scenario is a narrative description of what people do and experience as they try to make use of computer systems and applications. A scenario is an instance of a use case, that is, a use case specifies all possible scenarios for a given piece of functionality.

Routines: is a block of code that performs a single task. It is a building block of a computer program.

Global Data(Variable): global data(**Variable**) is data that is accessible in every scope or from every part of the software application.

Comment: is generally a written remark often related to an added piece of computer program to elaborate the intention of the program.

Module: A module is a collection of objects that are logically related. Those objects may include constants, data types, variables, and program units (e.g., functions, procedures, etc.). Note that objects in a module need not be physically related.

Coupling: in computer science, coupling or dependency is the degree to which each program module relies on each one of the other modules.

Cohesion: Cohesion measures the semantic strength of relationships between components within a functional unit.

Debugging: The process of locating and removing defects in a software system. Not to be confused with testing (a different activity).

Program Unit: A function, procedure, subroutine, process, main program, or other comparable object. Sometimes includes a compound statement (block).

Adaptive maintenance: modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment.

Perfective maintenance: modification of a software product performed after delivery to improve performance or maintainability.

Emergency maintenance: unscheduled corrective maintenance performed to keep a system operational.

Bug: computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program's source code or its design.

2.2 Policy authority

This Policy is the underlying guideline for any other subsequent documents that may come out as regulation, manual, procedures, etc. with regards to software development in Tsehay Bank. Core banking and data Analytic department of the Bank is authorized for implementation and interpretation of the software development policy.

CHAPTER THREE

3. Scope, purpose, objective and statement of the policy

3.1 Scope

This policy applies to all software development activities in the Bank from software development request or inception to software maintenance and up gradation activities after deployment. This document applies to:

- Software developers of the bank
- Team leaders and software development managers
- IT auditors and security department
- End users (i.e. departments of the bank)

3.2 Purpose

The purpose of the Tsehay Bank Software Development Policy is to describe the steps for developing and/or maintaining software in the Tsehay Bank environment, and to enforce standard and uniformity in the software development process.

3.3 Policy Objectives

The objectives of the policy are as follows:

- ✓ To set up the institutional framework to implement and monitor software development.
- ✓ To increase software quality and security.
- ✓ Improve the productivity of existing software developers.
- ✓ Allow new software developers to become comfortable with existing source code in less time than would otherwise be necessary.
- ✓ Allow existing software developers to move around to different projects easily without having to adjust to the development style in use by other groups.
- ✓ To increase the contribution of software products in the service delivery of the bank.

3.4 Policy statement

Several software development models exist to streamline the development process. Each one has its pros and cons, and it's up to the development team to adopt the most appropriate one for that specific project. Sometimes a combination of the models may be more suitable. Whatever type of model is used the following software development issues should be addressed as described below.

3.4.1 Software Project Initiation

Software project idea can be raised either from end users or from the software development team itself, or from other (third party) organizations. Since it has much better perspective in the area, the software development team is advised to be proactive and to watch for possible software projects.

3.4.2 Software Project Team

As soon as the initiated software project is approved to continue, software project team will be organized from existing human resource of the bank. Depending on the scope of the work the team can consist of only individuals from core banking and Analytic department or any other personnel of the bank.

3.4.3 Software Project Management

The software project team will have manager and he will be responsible for scheduling the task and managing the whole project. The project management task may vary depending on the scope of the project but to the minimum, the project schedule should be prepared and approved by the core banking and Analytic department director before going in to the next step. Developers must strictly follow standards in every phase of development.

3.3.4 End User Team

End user must have the responsibility starting from project initiation until project end to provide appropriate support for developer team.

3.3.5 Project proposal

A Project proposal is the initial framework for establishing the concept of the project and includes what you want to accomplish, an explanation of objectives, and plans for achieving them. It is common for a project proposal to include a list of activities or tasks that will be associated with the project, illustrate the significance of this specific project idea, and explain the origins of this project.

3.3.6 Requirement Analysis

The first task of the project team is identifying users' requirement. During requirements analysis the analyst gather system requirement from end users using different techniques and prepare requirement analysis document based on appendix x. This document must be approved by both the users and the software development manager before proceeding to the next phase of development. Client departments must provide all their requirements in a formal, clear and unambiguous way. They will be held accountable for inconveniences caused due to ambiguous and incomplete requirements or requirements that violate other business rules provided and approved by them.

3.3.7 Time Analysis

After the project requirement is finalized, the system Analysis must set the correct time flow of the project schedule.

3.3.8 System Design

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

The main objectives of the design are Practicality, Efficiency, Cost, Flexibility, and Security. Depending on the scope of the system to be developed and resources availability, there are different techniques of system design but the major outputs of System design activities are: Database design, Program design (which may include object diagrams), System Interface Specification, Audit Consideration. The system design document should consist at least the visual representation of the database, components of the system, and system interfaces. The system design document should get approval by the software development manager before proceeding to the next phase of development.

3.3.9 Coding

The goal of coding guidelines is to improve the productivity of all software development i.e. to make it easier, more reliable, and faster. While many coding styles are efficient and maintainable, agreeing on this set of guidelines allows the entire team to work cohesively. Close adherence to these guidelines is essential to producing software that is consistent project-wide and maintainable by diverse team members.

Unless and other wise for some convincing reason the coding of new system should be done using any of object oriented programming languages. In addition, the back end database shall be either any database management systems. Keeping this in mind the coding process should stick to the following points:

- ✓ Whenever you are coding ask yourself: “How will the next person understand that?”
- ✓ Consistency within a project is very important.
- ✓ There must be adequate input validation checks built into data entry programs.
- ✓ Passwords to all applications must be encrypted as per security team recommended.
- ✓ Use local repository server in the project.
- ✓ Programs with multiple files and multiple functions should be organized into modules. A module is a file or set of files that form a process or library function. All files in a module

should be in a single directory, except header files shared with other modules, which are in a common directory.

- ✓ All interfaces to modules should be through a well-defined set of operations.
- ✓ All comments should be high-quality comments that describe the actions of the surrounding code in a concise manner.

3.3.10 Coding convention

During development of the code, developers should follow coding conventions.

- ✓ Back end language must be **java and asp Mvc c#**.
- ✓ Framework must be latest **spring framework, and entity frame work**.
- ✓ Frontend language should be angular, react, vue as needed.

3.3.11 Testing

The purpose of software testing is to provide an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software and the functionality of the application developed. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs, so any feedback raised during testing should be seen from such mentality. The user tests the developed system and fills the form on appendix and approved the acceptance of the software.

3.3.12 Manual

The following manuals should be prepared: Installation and Configuration Manual, Systems Operations Manual, and User Manual. Each of these manuals should at least include the objective of the newly developed package and series of steps that should be followed to accomplish certain task.

3.3.13 Training

Training should be conducted for both end users and system support personnel by installing the package on test server. During training user feedback should be recorded along with test results for future reference.

3.3.14 Deployment

Software deployment is all of the activities that make a software system available for use. And during deployment, if there is any data conversion (from old to new system) it must be

audited, after deployment there must be library to store source code and update of any software products in use, each release should be given a version number. Updates or new versions should be put on server and pushed to clients to ensure whether the same software version is being implemented in all such user sites.

3.3.15 Maintenance

Software maintenance is the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment. The type of maintenance can be corrective, adaptive, perfective or emergency maintenance. The need for maintenance can be raised either from the user or the development team. Whoever raised it; the objective of the maintenance work should be clearly identified and documented before going deep in to it. Depending on the scope of the maintenance work, if the maintenance work is only little modification in certain part of the application and if its effect is not observable to end users the maintenance work can be done internally by the development team without notifying the end users, if the scope of the work requires modification of graphical user interface or sequence of execution of tasks, the maintenance work can be done internally by the development team and some explanation or very brief training on it can be conducted to users, If the scope of the maintenance work is wide and if it requires involvement of users it is better to follow the formal development process discussed before. In every of the above cases every piece of modification should be commented on the source code according to source code commenting procedures of the bank, and these changes should be reflected in every related documents.

CHAPTER FOUR

4. Applicability and use

4.1 Audience

The Tsehay Bank Software Development Policy applies equally to all individuals that participate in software development activities in the bank.

4.2 Acceptable Use

Tsehay Bank software development services are provided to Tsehay Bank's community and partly to public limited access primarily in support of its banking mission. Software Development Section of the bank encourages the use of the software resources and makes them widely available to Tsehay Bank community. Nonetheless, the use of these resources constitutes acceptance of this policy and is subject to the following limitations, necessary for the wellbeing of the overall software resources. The software resources should be used for the purpose for which they are intended. Software developers must respect the rights, privacy and property of the bank. Only registered software developers or those given permission by the designated authority are permitted to participate in Tsehay Bank software development process.

4.3 Violation

- ✓ Project is failed and the development is restart again.
- ✓ Violation of this policy may result an action by core banking and analytics department as per needed.

CHAPTER FIVE

6.1 Sample Template for Requirement Analysis Document and Use Case

1. Introduction

1.1 objectives

1.2 Scope of the project

2. Existing system

2.1 Introductions

2.2 Business process

2.3 problems existing system

3. Proposed system

3.1 Introductions

3.2 functional requirements using use case

Use Case Id
Use case name
Description
Actor
Pre-condition
Alternative course action
Post condition

4. Other Nonfunctional Requirements

4.1 Performance requirements

- 4.2 Safety requirements
- 4.3 Security requirements
- 2.4 Software quality attributes
- 4.5 Availability requirement:
- 4.6 Usability requirement
- 4.7 Accessibility requirement

Sample Template for Use Case Documentation

Use Case: Title

[There may be any number of use cases in a usage scenario.]

Issues

[List/describe/discuss the particular issues illustrated by this use case.]

Requirements

[Such things as performance requirements, e.g. fault tolerance, frequency, reliability.]

Actors/Roles

[Specific actors used in this use case (should be a subset of Scenario Actors).]

Goals/Context

[Goals of actors initiating processes; knowledge required to specify process inputs.]

Assumptions

[Conditions that must be true for successful use case execution and termination.]

Scenario/Steps

[Describe the steps of this use case, and specifying the actors/roles responsible for invoking and executing each step. Include, if relevant, information about process flows, protocols, messages, pre-conditions, post-conditions, errors, etc. associated with the steps. Describe interesting variants, where possible.]

Extensions

[Known issues not explicitly covered above (e.g., error handling).]

Ontologies and Semantic Descriptions

[Describe the kinds of ontologies and semantic descriptions that are required, rather than specific implementations.]

Reasoning Techniques Required

[Specific kinds of reasoning required, where known (e.g., planning, data collection or analysis, classification, similarity matching, rule-based inference).]

Annex



Software Development User Request form

Name:

Section:

Request Type: New _____ Modification _____

Request Description:

No	Description

Date:

Sign:

Dep. Manager Name:

Date:

Sign:

form1: request for

Sample Template for system testing

	List of the features to be tested	Input given (if any)	Expected output (or system requirement)	Test output	Remark
1					
2					

Form 2: testing form

Project Sign off template

1. Project Goal
Were project goals met?

2. Project Deliverables
Were project deliverables met?

3. Comments (If any)

4. Document Signatures			
Name	Description	Signature	Date
Service Owner Name	By signing this document, I acknowledge that I have received all the stated deliverables at the agreed to quality levels.		mm/dd/yyyy
Project Manager Name	By signing this document, I acknowledge that I have delivered all the stated deliverables at the agreed to quality levels.		
Others (if needed)	Add more rows if needed.		