



ፀሐይ ባንክ
TSEHAY BANK
ከህዝብ

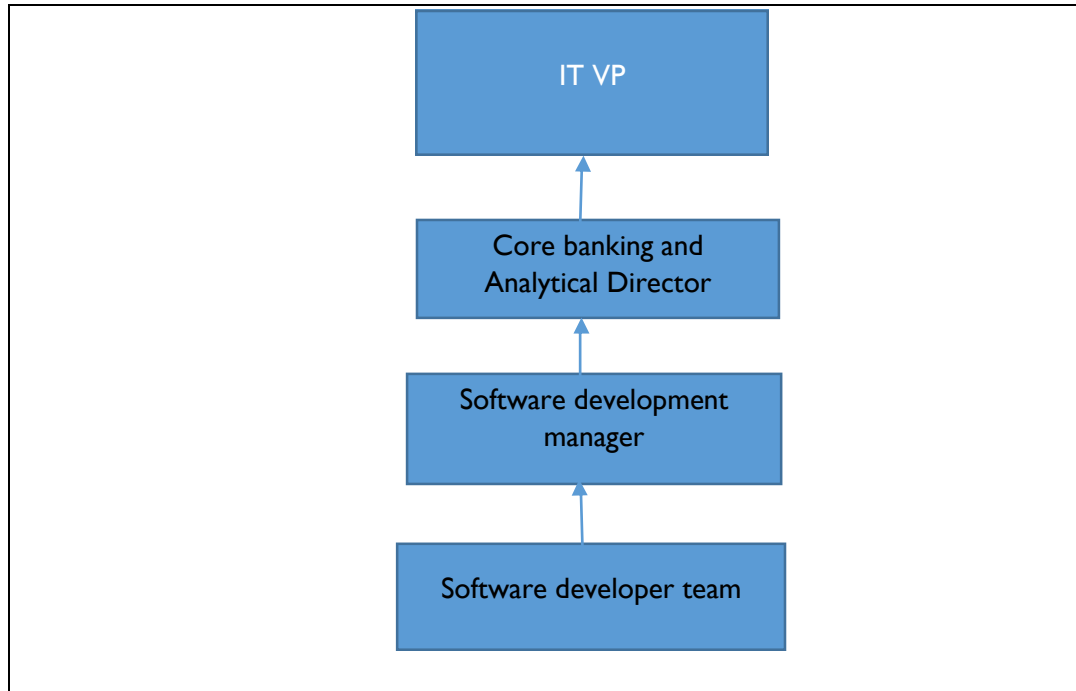
Software Development Procedure

(Draft Version 1.0)

1. Contents

| | |
|--|----|
| 2. Department structure | 2 |
| 3. Introduction: | 3 |
| 4. PURPOSE: | 3 |
| 5 Software Project Initiation procedure | 3 |
| 6 Responsibly Project team: | 3 |
| 6.1 Software developer Team: | 3 |
| 6.2 The Software Manager: | 3 |
| 6.3 End User Team | 3 |
| 7. Software Development Activities: | 4 |
| 7.1 Requirement Analysis phase | 4 |
| 7.2 Time Analysis | 5 |
| 7.3 Design Phase | 5 |
| 7.4 Implementation Phase (Coding Phase): | 5 |
| 7.5 Validation Phase (Testing): | 10 |
| 7.6 Installation Phase | 11 |
| 8. Maintenance Phase: | 13 |
| 8.1 The change management Process | 13 |

2. Department structure



3. Introduction:

Software Development Procedure is an established method of accomplishing software development, usually with steps that are followed in performing in a prescribed order in software development process.

4. PURPOSE:

This procedure provides an overall view of the activities of the software development life cycle and the processes associated with them. These software activities and processes support the development of software within the company framework. If a subcontractor participates in the development process or is outsourced, the subcontractor shall work according to this software development life-cycle procedure or according to an equivalent process. This document maps the various software development and support processes on the software development life cycle.

5 Software Project Initiation procedure

- ✓ Software project idea can be raised either from end users or from the software development team itself.
- ✓ Director of department requests software requests.
- ✓ Development department and requester department assign project-involved teams.

6 Responsibly Project team:

6.1 Software developer Team: has overall responsibility for all software developer.

- ✓ Communicate with software requester team.
- ✓ Requirement gathering up to software deployment activity.

6.2 The Software Manager: is responsible for on-going development activities and documentation of the software under development.

6.3 End User Team

End user must have the following responsibility starting from project initiation until project end.

- ✓ Assign involved client to project: this user must know the work of the department that the system perform because the input from the user is very important for development process.
- ✓ Client must provide proper information.
- ✓ Client must replay timely for any requests from developer team.

- ✓ 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

7. Software Development Activities:

The major software activities for a software development product include the following:

7.1 Requirement Analysis phase

After project team has agreed on the general development of the software, including its features and the services provided, request team and a SW development team will be involved in the process of.







- ✓ Gathering user requirement.
- ✓ Client departments must provide all their requirements in a formal, clear and un-ambiguous way.
- ✓ Writing Software Requirements Specifications document(SRS).
- ✓ Prepare presentation to project teams.
- ✓ The project team shall review the SRS document. Attempt shall be made to identify incomplete, ambiguous or conflicting requirements. In case of such requirements, the Software Manager shall be informed and shall be asked to reconcile such conflicts.
- ✓ Sign of requirements document before starting development.

Phase Output:

- a. Software Requirements Specification (SRS) approved document.
- b. Product schedule / Project plan verified and updated.

7.2 Time Analysis

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

| ID | Task Name | Start | Finish | Duration | Comple te | 2009/2/14 | | | | | | | | | | 2009/4/1 | | | | | | | | | | 2009/7/1 | | | | | | | | | | 2009/10/1 | | | | | | | | | | | | |
|----|--------------------------|-----------|-----------|----------|--------------|---|--------|---------|---------|--------|--------|---------|---------|---------|--------|----------|---------|---------|--------|---------|---------|---------|---------|--------|---------|----------|---------|--------|---------|---------|---------|--------|--------|---------|---------|-----------|--------|---------|---------|---------|---------|--|--|--|--|--|--|--|
| | | | | | | 2/2/09 | 2/9/09 | 2/16/09 | 2/23/09 | 3/1/09 | 3/8/09 | 3/15/09 | 3/22/09 | 3/29/09 | 4/5/09 | 4/12/09 | 4/19/09 | 4/26/09 | 5/3/09 | 5/10/09 | 5/17/09 | 5/24/09 | 5/31/09 | 6/7/09 | 6/14/09 | 6/21/09 | 6/28/09 | 7/5/09 | 7/12/09 | 7/19/09 | 7/26/09 | 8/2/09 | 8/9/09 | 8/16/09 | 8/23/09 | 8/30/09 | 9/6/09 | 9/13/09 | 9/20/09 | 9/27/09 | 10/4/09 | | | | | | | |
| 1 | Proposal writing | 2009/2/18 | 2009/2/22 | 5.0 d. | 100% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Requirement analysis | 2009/3/1 | 2009/4/30 | 61.0 d. | 0% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Design | 2009/4/1 | 2009/5/1 | 31.0 d. | 0% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Implementation | 2009/6/1 | 2009/9/30 | 122.0 d. | 0% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Installation and testing | 2009/10/1 | 2009/10/7 | 7.0 d. | 0% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Project close | 2009/10/8 | 2009/10/9 | 2.0 d. | 0% |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

7.3 Design Phase

Following the approval of the requirement specifications, review of the requirements and analysis of the system, the software development team shall initialize the design phase. The software design is comprised of top-level design and detailed design. The top-level design determines the software architecture, behavior, interfaces and the logical structure of the database. The detailed design defines the internal structures and interfaces.

7.4 Implementation Phase (Coding Phase):

Coding procedure used productivity of all software development i.e. to make it easier, more reliable and faster software. The purpose of this phase is to implement the SW requirements and verify that each program unit meets requirements as defined in the design documentation. During this phase, the following activities occur:

- ✓ Each team member is responsible to design, write, and modify his/her assigned segment(s) of the code under the supervision of the Software Manager.
- ✓ During the development of the modular segments of the code, each team member is also responsible to debug the designed code sequences. The debugging process is intended to ensure proper operation and data management and to identify any actual or potential operating problems or logic and syntax errors.

- ✓ Any errors or operating problems identified by the team members shall be then corrected through logical restructuring of the code segments. Additional debugging runs and repeated testing processes shall be conducted as needed by the team members following each code revision, until the code has no apparent errors.
- ✓ After the modular code segments have passed the code debugging, the code segments shall be integrated into one comprehensive sequence that is intended to synchronize all individual tasks identified in the requirements specifications.
- ✓ Unit tests shall be performed on all features after review to verify satisfaction of software requirements for safety related modules.
- ✓ The Software Team shall plan, perform and document the Module level testing in the safety related modules before being transferred to integration for other software engineers to use. The functions to be tested are listed in a checklist that has been prepared before hand. The functions found on the checklist that were tested and not tested are to be summarized.
- ✓ All software shall be under revision control as defined in the QA section. The test results are filed with the Quality Records.
- ✓ The Software shall initiate code reviews for any potential risk function or complex function.
- ✓ 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 and the encryption type is as per security team recommendation.
- ✓ 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.
- ✓ Use proper versioning.
- ✓ Programing language must be **java and asp Mvc c#**.

- ✓ Framework should be latest **spring framework and entity frame work**.
- ✓ Frontend should be angular, react, vue as needed.

Module Header Comments

A module header is a block of comments at the beginning of a source file. Module headers should look like the following:

```
/******  
  
/*Copyright(c) Tsehay Bank S.C.          */  
  
/* All Rights Reserved                    */  
  
/* An Unpublished Work                   */  
  
/*                                     */  
  
/* This is a Proprietary program product material */  
  
/*and is the property of Tsehay Bank S.C.      */  
  
/* No sale, reproduction or other use of this */  
  
/* program product is authorized except as granted */  
  
/* by the fully executed Tsehay Bank SC product */  
  
/* license or by the separate written agreement */  
  
/* and approval of Tsehay Bank S.C.          */  
  
/******  
  
/* Author: XXXX YYYY                      */  
  
/* Revision History:                       */
```



```
/* << date and explanation of each change >>      */

/***** */

/* Version 22.1                                     */

/* Checked in 1/24/96 at 12:39:03                   */

/* Retrieved 4/22/96 at 18:16:48                     */

/***** */

/* Purpose:                                         */

/* -----

/* This file contains the code to generate a string */

/* value containing a specified number of

occurrences /* of a specified string

*/


/***** */
```

Coding convention

- ✓ All identifiers that represent words or phrases must be English words or phrases.
- ✓ Never use a numeric suffix to differentiate two names.
- ✓ Do not use names with similar meanings for different objects in your programs.
Do not use similar names that have different meanings.
- ✓ Avoid misspelled words and names that are often misspelled in identifiers.
- ✓ Declare variables as locally as possible. And always initialize variables.
- ✓ Declare variables as close as possible to where it is first used.
- ✓ Use one variable declaration per line.

- ✓ Avoid using member variables. Declare local variables wherever necessary and pass it to other methods instead of sharing a member variable between methods. If you share a member variable between methods, it will be difficult to track which method changed the value and when.
- ✓ Never hardcode a path or drive name in code. Get the application path programmatically and use relative path.
- ✓ Show short and friendly message to the user. But log the actual error with all possible information. This will help a lot in diagnosing problems.
- ✓ Avoid passing too many parameters to a method. If you have more than 4~5 parameters, it is a good candidate to define a class or structure.
- ✓ A method should do only 'one job'. Do not combine more than one job in a single method, even if those jobs are very small.
- ✓ Do not use variable names that resemble keywords.
- ✓ Avoid using abbreviations. When necessary, use standardized abbreviations or ask someone to review your abbreviations. Whenever you use abbreviations in your programs, create a "data dictionary" in the comments near the names' definition that provides a full name and description for your abbreviation.
- ✓ If the variable is used only as a counter for iteration and is not used anywhere else in the loop, many people still like to use a single char variable (I) instead of inventing a different suitable name.

Repository convention

- ✓ The project name should be descriptive and clear.
 - ✓ Do not git push to straight to the master branch, use merge request.
 - ✓ Do not commit code as unrecognized author.
 - ✓ Do not leak secrets in to the repository.
 - ✓ Do not commit dependencies in to the repository.
 - ✓ Do not commit local config files in to the repository.
 - ✓ Create a meaning full git ignore file.
 - ✓ Use branch-naming conventions.
 - ✓ Use proper commit naming.
 - ✓ Use similar change in the same commit.
- 

API Integration


The bank will integrate with third party using API, with integration process the following requirement should consider.

Two types of integration

Third party with bank (incoming Integration)

- For incoming integration there should be authentication for third party to secure the integration.
- Incoming integration must be XML or JSON.
- Incoming integration should have constant JSON or XML request and response format as per the integration type.
- The bank should share API specification document to third party.
- Validate request is valid or invalid.
- Log request and response on daily and company based on file and database for error fixing.
- Route to appropriate company API based on customer need to enquiry or pay.

Bank with third party (outgoing integration)

- Bank must accept API specification document.
 - Integration should be authentication for third party to secure the integration.
 - Log xml request and response on daily and company based on file and database for error fixing.
- 

7.5 Validation Phase (Testing):

The purposes of this phase is testing of software using different testing methodology like integration other software units within the system. Testing should include Functional and Non-functional tests. Functional tests (also known as specification-based, or “black box”) are defined as executing test cases from the requirement and functionality point of view and requester department must prepare test case document.

7.6 Installation Phase

After the completion of project validation to perform the initial installation. After the installation is performed, a trained tester shall check to verify that the installation output is correct. Once the software uploaded to the target device is verified, an Approval meeting is held by required participant's and project owner team. Software approval indicates that:

- a. Software verification and validation processes have been completed.
- b. A formal review was conducted.
- c. All safety/efficacy related issues reviewed and approved.

Phase Output: Software Approval Form with known Bugs list and sign off



Tsehay bank project Sign form

| | |
|--|---|
| Project Name: Payroll Management System | Project Manager: Name |
| Start Date: | Completion Date: |
| Project Duration: 3 months | Sponsor: |
| Project Goal: develop automate Payroll Management System management for finance department. | |
| Project Deliverables: the project is Deliverables to finance department that operate payroll. | |
| Clients: | |
| By signing this document, I acknowledge that I have delivered all the stated deliverables at the agreed to quality levels. | By signing this document, I acknowledge that I have received all the stated deliverables at the agreed to quality levels. |
| Project Manager Name and Signature: | Sponsor Name and Signature: |
| Date: | Date: |
| Remarks | |

8. Maintenance Phase:

The maintenance activities may include all engineering phases previously defined, from requirements to system test, to be performed according to Modification characteristics. All changes in the Software shall be handling with the Change Control procedure. Requested changes/modifications shall be transferred to the authority list and Software Manager who are required to select one of the following options:

- a. Reject the change/modification request,
- b. Approve the change/modification request for the current version, or
- c. Accept the change/modification request for later versions.

8.1 The change management Process

The primary goal of the Software change management is to accomplish software changes in the most efficient manner while minimizing the business impact, costs, and risks. All software changes within the Tsehay bank will be documented in the company's selected technology platform. To achieve this, the change management process includes the following primary steps (note that all information collected in the steps below is documented in a change record created in the Tsehay bank's selected technology platform).

- **Formally Request a change.** Requesting body fills and submits Request for change form (CRF) to software development department. All requests for change will be documented within the Tsehay bank's selected technology platform by creating a new change record.
- **Assign Change coordinator:** Department of software development assigns change request coordinator.
- **Categorize and prioritize the change.** The change coordinator will assess the urgency and the impact of the change on the infrastructure, end-user productivity and budget.
- **Analyze and justify the change:** The change coordinator works with the change requester and the change initiator to develop specific justification for the change and to identify how the change may affect the infrastructure, business operations, and budget. The change coordinators use this information to further research and develop risk and impact analysis. When completing of the change, the change coordinator must ensure they consider the business as well as the technical impacts and risks.
- **Approve the change:** The change coordinator submits the request for change document to Tsehay bank software development director for approval.
- **Plan and complete the implementation of the change:** This process includes developing the technical requirements, reviewing the specific implementation steps, design and develop the requested change and then completing the change in a manner that will minimize the impact on the infrastructure and end-users.
- **Post-Implementation Review:** A post-implementation review is conducted to ensure whether the change has achieved the desired goals. Post-implementation actions include deciding to accept, modify or back-out the change; contacting the end-user to validate

success; and finalizing the change documentation within the company's selected technology platform.

| <Project Name and/or Logo> | | Project Change Request (CR) | |
|--|--|-----------------------------|---------------------------|
| Change Request Submission Department | | | |
| Change Request Title: <i>Addition of one full-time (FT) database administrator</i> | | | CR Number: (CRC Use Only) |
| Change Request Category | <input type="checkbox"/> – Scope <input type="checkbox"/> – Time <input checked="" type="checkbox"/> – Cost | | |
| Originator Name | [Name] | Originator Organization | [Organization name] |
| Date Submitted | [XX/XX/XXXX] | Originator's Manager | [Manager name] |
| Primary Contact Person | [Name] | Backup Contact Person | [Name] |
| Primary Contact E-mail | [E-mail] | Backup Contact E-mail | [E-Mail] |
| Priority: (Check One): <input checked="" type="checkbox"/> 1 – Critical: Work stoppage or severe impact on productivity has occurred; solution needed immediately. <input type="checkbox"/> 2 – High: Work stoppage or severe impact on productivity is eminent; solution needed before impact occurs. <input type="checkbox"/> 3 – Medium: Impact on productivity is expected; workaround has been identified and solution is needed. <input type="checkbox"/> 4 – Low: Impact on productivity is minimal; solution is needed. | | | |
| Detailed Description of Proposed Change | <i>Data conversion is taking more time than planned due to the condition of the source data. Data conversion tasks are on the critical path and are at risk of slippage. Adding another full-time database administrator will add needed capacity to stay on schedule.</i> | | |
| Justification for Change | <i>Change will prevent impact on project end date due to slippage of critical path data conversion tasks.</i> | | |
| Current Workaround (if applicable) | None | | |

| | | | | | |
|--|--|--------------|----------------|--------------------|----------------|
| Potential Cost Considerations (if known) | Staffing costs will increase by 1 full-time database administrator. | | | | |
| Additional Information / Comments | None | | | | |
| Change Request Disposition (CRC Use Only) | <input type="checkbox"/> Rework <input type="checkbox"/> Not Accepted <input type="checkbox"/> Withdrawn <input type="checkbox"/> Deferred <input checked="" type="checkbox"/> Approved for Implementation Disposition Comments: Signature: _____ Date: _____ | | | | |
| Change Request Analysis Section | | | | | |
| CR Analyst (CRA) | [CR Analyst Name] | | | | |
| Date CRA Assigned | XX/XX/XXX X | Date CRA Due | XX/XX/XXX X | Date CRA Completed | XX/XX/XX XX |
| Impact Areas: (Check One or More) <input checked="" type="checkbox"/> Project Cost Increase: Change will result in an increase to project costs. <input type="checkbox"/> Project Cost Decrease: Change will result in a decrease to project costs. <input type="checkbox"/> Scope Increase: Change will result in an increase to project scope of work. <input type="checkbox"/> Scope Decrease: Change will result in a decrease to project scope of work. <input type="checkbox"/> Schedule Change: Change will result in a change to the Master Project Schedule (MPS). | | | | | |
| How much was increased or decreased in Cost, Scope, or Schedule | The project cost increased by 10%. | | | | |
| Solution Description and Impact on Baseline Items and the Project | Recruit and hire one additional FT DBA, baseline budget will increase. | | | | |

| | |
|---|---|
| | |
| Implementation Risks and Mitigation | None |
| Alternative Solutions Considered | <ul style="list-style-type: none"> • Extending data conversion schedule – rejected because extending the data conversion schedule will cause the project to miss the mandatory implementation date • Retain contractor staff – rejected due to the length of time procurement of external resources will take |
| Additional Information/Comments | None |
| Change Request Approval Section | |
| Change Analysis Reviewed (Name and Initials of representative from each team who reviewed CR Analysis) | Print Name: [Name] Initials: [Initials] |
| | Print Name: [Name] Initials: [Initials] |
| | Print Name: [Name] Initials: [Initials] |
| | Print Name: [Name] Initials: [Initials] |
| Change Request Disposition (CRC Use Only) | <input type="checkbox"/> Rework <input type="checkbox"/> Not Accepted <input type="checkbox"/> Withdrawn <input type="checkbox"/> Deferred <input checked="" type="checkbox"/> Approved for Implementation Disposition Comments: Signature: _____ Date: _____ |
| Change Request Tracking Section | |
| Changes Implemented | Hired 1 additional FT DBA |

| | |
|--|---|
| (Change Analysis / Implementation Team Use Only) | |
| Changes Verified (Change Analysis / Implementation Team Use Only) | <i>New DBA hire verified XX/XX/XXXX</i> |
| Change Request Closing Section | |
| Change Request Disposition (CRC Use Only) | <input type="checkbox"/> Closed Disposition Comments: Signature: _____ Date: _____ |