Paprika Financial App

Configuration Management Plan

Team Paprika
Paprika Financial App
Andrew Feltham
2018-03-08
Version 1.1

____
[SCM Lead Signature Block]

____
[Project Manager Signature Block]

Revision Sheet

Revision	Date	Brief Summary of Changes
Version 1.0(draft)	2018-02-19	Baseline document draft
Version 1.1	2018-03-08	Baseline Change diagram, explanation of branches (3.2.1)

1. INTRODUCTION

1.1 Purpose

This document provides information for configuration management for the Paprika Financial App. This document is intended for all members of Team Paprika.

1.2 Scope

This document defines Configuration Management (CM) activities for all Software and Data produced during the development of the Paprika Financial App (PFA) software. This document applies to all module products, end-user products, and data developed and maintained for the PFA. CM activities as defined herein will be applied to all future PFA projects.

This document conforms to IEEE standards for software configuration management and will change as needed to maintain conformance

1.3 Definitions and Acronyms

1.3.1 Key acronyms

BCR -	Baseline Change Request
CCB -	Configuration Control Board
CER -	Change/Enhancement Request
CRC -	Configuration Request Coordinator
CM -	Configuration Management
CRC -	Change Request Coordinator
DT&E -	Developmental Test & Evaluation
FOT&E -	Final Operational Test and Evaluation
PFA -	Paprika Financial App
PM -	Program Manager
SC -	Software Configuration
SCM -	Software Configuration Management
SCR -	Software Change Request
SE -	Software Engineer
SPR -	Software Problem Report
SQA -	Software Quality Assurance

1.3.2 Key terms

configuration management “A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics of a configuration item, control changes to those characteristics, record and report change processing implementation status and verify compliance with specified requirements.” [IEEE90].

customer “The individual or organization that specifies and accepts the project deliverables. The customer may be internal or external to the parent organization of the project, and may or may not be the end user of the software product. A financial transaction between the customer and developer is not necessarily implied.” [IEEE87]

database “A collection of data fundamental to a system.” [IEEE91]

installation “The period of time in the software life cycle during which a software product is integrated into its operational environment and tested in this environment to ensure that it performs as required.” [IEEE91]

plan “A detailed scheme, program, or method worked out beforehand for the accomplishment of an objective.” [Heritage85]

process “A sequence of steps performed for a given purpose.” [IEEE90]

project -- unit of work to meet a specific customer requirement. Includes all tasks, activities, and functions necessary to meet the requirements.

project deliverables “The work product(s) to be delivered to the customer. The quantities, delivery dates, and delivery locations are specified in the project agreement.” [IEEE87]

quality assurance “(1) A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.” [IEEE90]

“(2) A set of activities designed to evaluate the process by which products are developed or manufactured.” [IEEE90]

review --A process or meeting during which a work product, or set of work products, is presented to program personnel, managers, users, customers, or other interested parties for comment or approval.

Types include requirements review, design review, code review, test readiness review, formal qualification review.

software “Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system.” [IEEE90]

software “The application of a systematic, disciplined, quantifiable
approach
engineering to the development, operation, and maintenance of software; that is, the application of engineering to software.” [IEEE90]

software life cycle “A project-specific, sequenced mapping of activities.” [IEEE91]

software quality assurance --See quality assurance.

specification “A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and, often, the procedures for determining whether these provisions have been satisfied.” [IEEE90]

walk-throughs “A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code and the participants ask questions and make comments about possible errors, violations of development standards, and other problems.” [IEEE90]

work product --Any tangible item that results from a project function, activity, or task. Examples of work products include customer requirements, project plan, design documents, source and object code, user’s manuals.

1.4 References

IEEE87 IEEE Std 1058.1-1987, IEEE Standard for Software Project Management Plans

IEEE88 IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary.

IEEE90	IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (ANSI).
IEEE91	IEEE Std 1074-1991, IEEE Standard for Developing Software Life Cycle Processes.
Schach96	Classical and Object Oriented Software Engineering, Schach, 1996
Davis90	Davis, A, "Software requirements: Analysis & Specification," Prentice Hall, 1990.
Heritage85	The American Heritage Dictionary, Houghton Mifflin Publishers, 1985.
Paulk93	SEI Capability Maturity Model, Version 1.1, CMU/SEI-93-TR-24.
MIL498	MIL-STD-498, Military Standard Software Development and Documentation, 5 December 1994.

2. Software Configuration Management (SCM)

2.1 SCM Organization

Figure 3.1 provides an overview of PFA SCM organization.

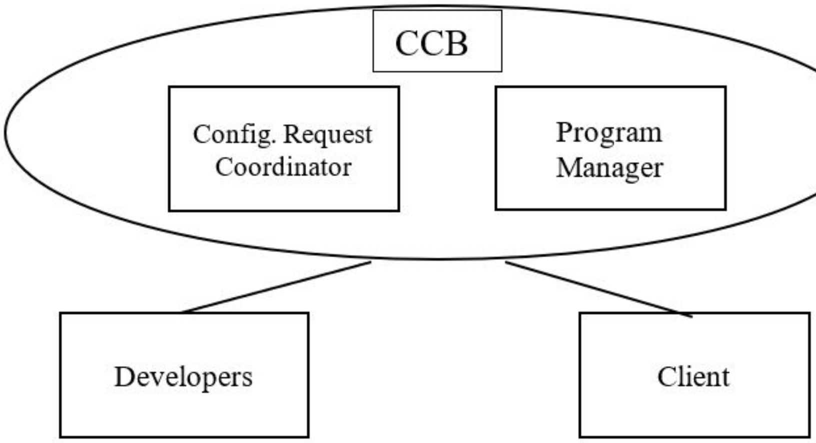

Figure 3.1. [Project Abbreviation] SCM Organization

2.2 SCM Responsibilities

Program Manager. The Program Manager (PM) is responsible for ensuring the SCM process is developed, maintained and implemented. The PM is responsible for ensuring that project leads and team members are adequately trained in SCM policy and procedure. The PFA Program Manager and Project Leads work together to develop, maintain, and implement effective software configuration management. Please refer to Figure 3.1.

The PFA PM has the authority to ensure development of an appropriate configuration management process. Since CCB responsibility exists with the customer (See Figure 3.1), team member(s) of the PFA Project will be identified as members of this CCB and will represent our interests there.

Configuration Control Board. The CCB is responsible for the review and approval of all Software Change Requests (SCRs), all baseline items, and all changes to baseline items. The CCB considers the cost and impact of all proposed changes and considers the impact to all interfacing items as part of their approval action. The CCB for PFA consists of the PM and SCM.

The associated Configuration Control Board (CCB) has the authority to approve a baseline version of this process, as well as any changes to the process as recommended by any associated CCB representative. The same CCB has the authority to define the baseline for each software module, and end-user products, and to approve start of work and acceptance of completed changes to the baseline products. This CCB also has the authority to approve reversion to a previous version of the product if warranted.

Software Configuration Manager. The PFA Program SCM is responsible for implementing the actions of this process and for the implementation of Team Paprika process improvement recommendations.

The Software Configuration Manager (SCM) ensures that the process is implemented once approved, and that proposed changes to the process are reviewed by PFA team members and approved by the CCB as required.

2.3 SCM Resources

GitHub and Slack should be used for SCM activities. The CBC should allow for two hours per week for SCM activities and should utilize additional time as CER’s occur.

3. SCM ACTIVITIES

3.1 Configuration Identification

3.1.1 Specification Identification

All changes to PFA code and documentation are to be approved by the CCB. While PFA remains in development the SCM will maintain the following baselines:

- source files
- executable files
- database
- libraries and frameworks
- unit tests
- design documents
- design diagrams

3.1.2 Change Control Form Identification

Software Change Request (SCR) procedures for the development team are defined in Appendix A. These procedures are used to add to, change, or remove items from the baselines. The identification and tracking of change requests is accomplished through Change/Enhancement Requests (CER).

3.1.3 Library

PFA utilizes GitHub for its software configuration management. All documentation and code are kept in version control using the “paprika” repository.

3.2 Configuration Control

3.2.1 Procedures for processing change requests and approvals

The master branch is used for release versions only. The development branch is a child of master and is used during sprints. All other branches are child branches of development. All developers are responsible for 1-3 child branches. Each branch is used for either general purposes or a specific app function. Branches for functions are deleted after they pulled into development. All requests for changes to baseline items within each project follows the Software Change Request procedures shown in Appendix A. These procedures are outlined below for PFA maintenance and production lifecycles:

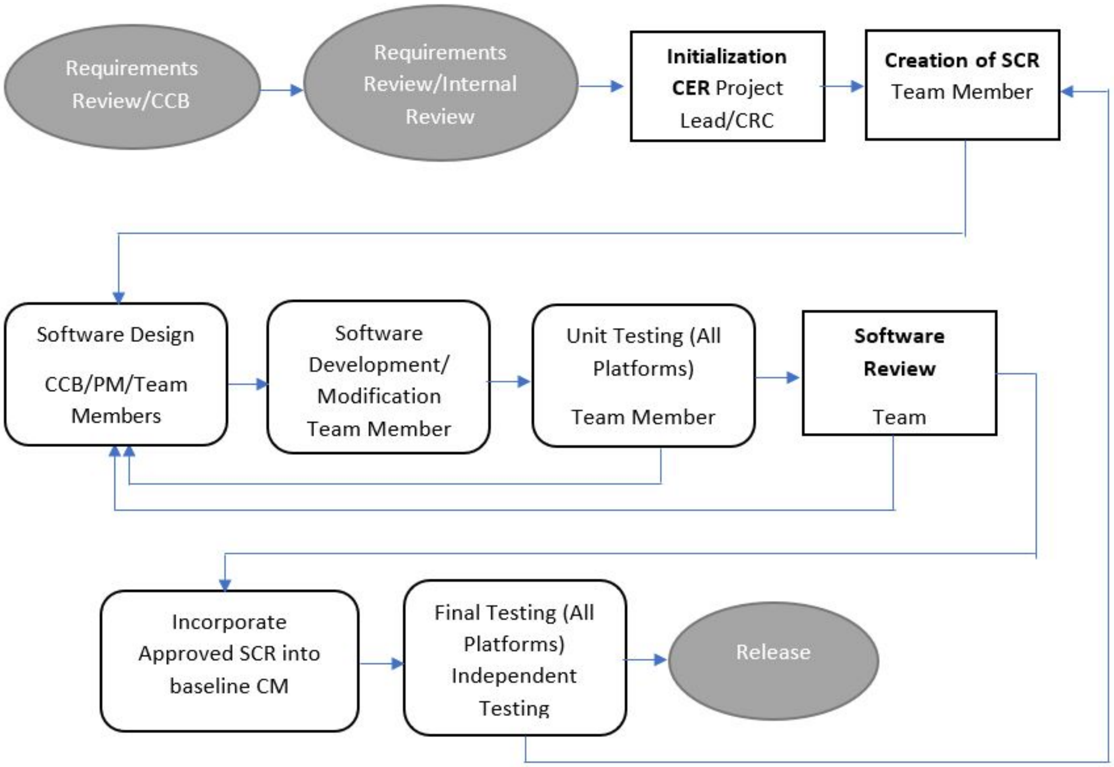
Requirements Review/CCB
Requirements Review/Internal Review
Initialization CER Project Lead/CRC
Creation of SCR Team Member
Software Design CCB/PM/Team Members
Software Development/ Modification Team Member
Unit Testing (All Platforms) Team Member
Software Review Team
Incorporate Approved SCR into baseline CM
Final Testing (All Platforms) Independent Testing
Release

Figure 4-1 Software Baseline Change Process

3.2.2 Change Control Boards (CCBs)

The role of the CCB is described in section 2.2. The CCB does not have any additional documentation.

3.2.3 Level of control

Changes to existing software baselines are evaluated by the software product's associated CCB. Changes to existing software baselines are either approved or disapproved by the associated project CCB. Approved changes are incorporated into the product baseline by the development team as identified in Appendix A.

3.2.4 Document revisions

The CRC is responsible for monitoring the PFA SCM. Updates will be made as any change occurs, and will be evaluated by the PM. Changed versions will be placed into the Team Paprika Google Drive folder, and team members shall be notified through Slack.

3.2.5 Storage, handling and release of project media

The initial approved version of any PFA software product is separately maintained by the CRC until the approval of subsequent version of the product. Upon approval of the subsequent version the initial version is held in the project SCM library. This library is under configuration control.

3.2.6 Reporting

The status report of requested changes is available to the PM and development team at any time. Reports will be prepared by the CRC.

3.2.7 Release process

PFA will have a release version created at the end of each sprint. A release of PFA requires the approval of the PM. The PM will select the version of each baseline, and the CRC will ensure the release branch is configured accordingly.

4. Training

Team Paprika does not have a required SCM training plan. Team members are expected to keep up to date with the software control process.

APPENDIX A // Software Change Request Procedures

A change to PFA software may be requested by a Software Problem Report (SPR) or Baseline Change Request (BCR). A change to PFA documentation may be requested by a Document Change Request (DCR). All the above requested change procedures are referred to as a change enhance request (CER) in the text below.

The Configuration Control Board (CCB) for each development team determines which of the change requests is required for each software release prior to the start of work for that release. Additional change requests are reviewed by PFA CCB to determine and assign the proper status to the change requests, these are held for CCB scheduling. Status is one of the following:

Open - Change request is to be implemented for current software version.
Hold - Change request targeted for another software version.
Voided - Change request is a duplicate of an existing CER or does not apply to existing software.
Working - Change is currently being implemented.
Testing - Implemented change request is under DT&E evaluation.
Fixed - Change request is implemented, unit and integration test complete.

The following steps define the procedure for each status. Each procedure starts with the change request being given to the Project Manager or designated Change Request Coordinator (CRC).

A. Open

1. INITIALIZATION

a) The CRC for the associated PFA project creates a CER using the CER system.

b) The CRC notifies the Project Manager of the new CER.

2. CREATE SCR

a) The PM reviews the CER and provides estimates of schedule impact to CCB.

b) CCB prioritizes CER and authorizes implementation.

c) The PM assigns the CER to a Software Engineer (SE) to incorporate.

d) The SE creates a Software Change Request (SCR).

3. SOFTWARE DESIGN

The SE designs changes necessary to implement CER:

a) The SE identifies interface changes and consults with appropriate Pl.

b) Determines changes needed.

c) The SE identifies any changes needed for documentation.

d) The SE estimates work effort required for CER completion.

4. SOFTWARE DEVELOPMENT/MODIFICATION

b) The SE makes needed changes and provides updated documentation for testing.

c) The SE checks out the files from the configuration managed version (not the baselined version) of all the files needed to incorporate the CER.

NOTE: PFA utilizes GitHub for configuration management. GitHub keeps track of the product release versions of each source file and is used to check-in and checkout files. This product allows version independence, keeping each released version intact, allowing for release backdating. The repository for PFA is paprika.

d) The SE fills in the CM sections of the SCR.

e) The SE puts the changes in the configuration managed files.

5. UNIT TESTING

The SE tests on all platforms, if test fails go to Step 3, Software Design.

6. SOFTWARE REVIEW

a) The SE prepares a software review package:
1) Hard-copy of SCR.
2) List of documentation changes.

b) Software Review

1) SE gives software review package to another team member for review.
2) If SCR peer approved go to step 7.
3) SCR not approved go to step 3, Software Design.

7. INCORPORATE APPROVED SCR INTO BASELINE CM

a) The PM changes the SCR status to indicate that it is approved.

b) The SE checks in the files.

c) The SE fills in the CM sections of the SCR.

f) The CRC reviews SCR for completeness.

g) The CRC changes the status on the CER to indicate testing.

h) In the case of significant changes, the CRC creates a list of changed files for the development team.

i) The designated SCM installs changes to baseline on all platforms.

8. FINAL TESTING

a) The CRC identifies all CERs with testing status.

b) Final testing by DT&E validation group (If problems are found start over at step 2a)

9. CLOSURE

The CRC changes final status on CER to indicate complete.

B. Hold

The CRC creates a CER using the CER system on the SUN marking status HOLD and indicates the targeted version for incorporating the change. The CRC files the original change request for the next version. The change will be considered for inclusion in the next version.

C. Voided

1. Duplicates:

a) The CRC marks the change requests as a duplicate, specifying which CER is a duplicated.

b) The CRC files the change request for associated version.

2. Other:

a) The CRC marks the reason for cancellation.

b) The CRC files the change request for associated version.

D. Testing

The CRC forwards CER/SCR to PM for review. After review, the PM forwards CER to a SE to test for validation. CER status is updated to indicate testing status.

E. Fixed

The CRC files the closed change request and updates database.

APPENDIX B // PFA Software Organization

G.1 PFA Organization

PFA software elements can be categorized into three groups: module products, data products, and end-user products.

Module products and data products are used by software developers and not the users and are components of PFA.

Module products - Module products are software libraries that perform a specific set of functions. Module products provide re-usable code ensuring consistency in function performance, eliminating duplication of effort in developing like functions, and reducing the amount of code. Module products are used in our end-user products and may be available for use by other companies/agencies.

Data products - Data products are database data files. Data products allow a single development, maintenance, and testing source and provides consistent data and format. Data products are included as part of our end-user products and may be available for use by other companies/agencies. Data products may be sent to users as a product itself.

End-user products - End-user products are executable programs for users.

Configuration management of module products is accomplished according to this document. Each module product has an individual baseline and version numbers. Module product managers specify the version of the module product to be used in end-user products.

G.1.1 PFA Module Products

Currently no module products.

G.1.2 PFA Data Products

Currently no data products.

G.1.3 PFA Deliverables

Currently no deliverables.

G.1.4 PFA Products

PFA product lists will be maintained and updated as they are created. This is the responsibility of the SC manager.