Software Configuration Management Plan

Avian Limited - Java Air

Amy Erxleben

November 7, 2016

Approvals

Title	Signature	Date
Integration Engineer	Amy Erxleben	10/03/16
Integration Engineer	Amy Erxleben	10/05/16
Integration Engineer	Amy Erxleben	11/04/16

Revision History

Version	Author	Modifications	Date
0.1.0	Erxleben	Created first draft	10/01/16
0.2.0	Erxleben	Added sections "Approvals", "Revision History", and "Annex A", added content to sections 3.2.2.1-3.2.4	10/03/16
1.0.0	Erxleben	addded content to sections 3.3.2.1, 3.4, 3.6. Updated content in section 3.2.2.1 & 3.3.6, modified format of Approvals and Revision History	10/05/16
2.0.0	Erxleben	sections 3.3.2.1 modified, Added Annex A, backup plan	10/31/16
2.1.0	Erxleben	updated sections 3.2.3 and 3.3.2.1	11/04/16
2.1.1	Erxleben	corrected typos	11/07/16

Contents

3.1 Introduction

This Software Configuration Management Plan (SCMP) describes how the artifacts for the Java Air ticket booking project are to be managed.

3.1.1 Definitions

Approved CIs: CIs that have been approved by the project management
Artifact: Any product of the project (e.g document, source code, object code, test result)
Master file: A particular file designated for this project to describe version history, described in section 3.3.1.2

3.1.2 Acronyms

CI: configuration item-an item (artifact) tracked by the configuration system
CM: configuration management-the process of maintaining the relevent versions of the project
SCMP: the Software Configuration Management Plan (this document)
SDD: the Software Design Document
SPMP: the Software Project Management Plan
SQAP: the Software Quality Assurance Plan
SRS: the Software Requirements Specification document
STD: the Software Test Documentation
STP: the Software Test Plan
SVVP: the Software Verification and Validation Plan

3.2 SCM management

3.2.1 Organization

A "Configuration Leader" will be designated from the engineering team for the duration of the project.

3.2.2 SCM responsibilities

3.2.2.1 Configuration Leader

The configuration leader will be responsible for organizing and managing configuration management (CM). CM plans should be discussed with the development team prior to implementation whenever possible. The configuration Leader will maintain this document (the SCMP). Installation and Maintenance of the configuration management tools specified in section 3.2.3 of this document is also the responsibility of the configuration leader. A backup plan for configuration management will be developed by the CM leader to be used in the case that the CM tools become discontinued, no longer supported, or malfunction. Additional responsibilities are stated in sections 3.1-3.6.

3.2.2.2 Project Leader

If the configuration leader is unable to perform his/her duties due to extreme circumstances, the project leader will take over these responsibilities. The project leader is responsible for knowing how to access all documents for the duration of the project. The project leader is also responsible for ensuring that archiving is performed in accordance with the policies in section 3.2.3 below. Additional responsibilites of the project leader are stated in sections 3.3.3 and 3.3.4.

3.2.2.3 Engineers

Engineers are responsible for abiding by the policies published in the most current version of this document. Additional responsibilities of the engineers are stated in section3.3 below.

3.2.3 Applicable policies, directives, and procedures

1. All current and previously released versions of CIs will be retained.
2. The master file (defined in section 3.3.1.2) can only be accessed by the configuration leader or the project manager. The project manager should only access this file if the configuation leader is absent.
3. The preffered development tool is NetBeans IDE 8.2.
4. Java Air will use GitHub in conjuction with TortoiseGit as a configuation management tool.

3.2.4 Management of the SCM process

3.3 SCM activities

3.3.1 Configuration identification

3.3.1.1 Identifying configuration items

The project leader is responsible for identifying all CIs. Engineers may propose new CIs to the project leader. The project leader's approval is required before a new CI is created. Approval must be documented via email correspondance or team meeting minnutes. If the project leader has not responded to a CI request within one day, the engineer may gain approval from the configuration leader.

3.3.1.2 Naming configuration items

The configuration leader has the responsibility of naming all CIs. The file conventions are as follows:

Root directory: JavaAir

File N_N_N.xxx corresponding to version N.N.N

For example version2.5.9 of the SDD will be found at JavaAir/SDD/2_5_9.doc

A text file named "Master" located in the root directory will keep a record of the current and past states of the project. Some examples of what this file should include are:

> The current version of Java Air ticket booking application is 3.2.2. This includes version 2.2.1 of the SRS, and version 1.5.9 of the SDD.
>
> The previous version of Java Air ticket booking application is 3.1.9. This included version 2.2.1 of the SRS, and version 1.4.0 of the SDD.

This information will be maintained in a table with headings: Java Air ticket booking version, SRS version, SDD version, . . . , Release

3.3.1.3 Acquiring configuration items

Engineers who require CIs to make changes will check them out using GitHub in conjuction with TortoiseGit. Engineers should create and designate a folder on their workstation as a clone of the GitHub repository. Each time an engineer wishes to modify a file, a "pull" action should be berformed to ensure the most recent versions of all files are on their local drive. Under no circumstances may an engineer transfer a CI directly to anyone.

3.3.2 Configuration control

3.3.2.1 Requesting changes

Every request for change to a CI must include the following: The name and version of the CI in which the change is being requested, the date of the request, the name of the engineer(s) making the request, a description of the change requested and explaination of need for the change. Additional information may be included in the request to clarify the significance and to assist in the evaluation of the request.

3.3.2.2 Evaluating changes

The project leader will evaluate all proposed changes and must also specify the required quality standards for incorporation.

3.3.2.3 Approving or disapproving changes

The project leader must approve all changes. If the project leader is unreachable for two days afer the submsission of a proposed change, the configuration leader may apporve the change.

3.3.2.4 Implementing changes

The configuration leader will be responsible for coordinating the testing and integration of any approved CI changes. This should be done in accordance with the process and standards described in the Software Test Documentation. The configuration leader is responsible for coordination of the building of a version for testing. Version releases must be cleared with the project leader.

3.3.3 Configuration status accounting

The configuration leader will update the configuration summary at least once a week on the project's configuration.

3.3.4 Configuration evaluation and reviews

The project manager shall schedule a review by the CM leader of the project configuration at least once every two weeks. This will be included in the agenda of the week's regularly scheduled meeting. At this time, the CM leader will review the configuration management status and report proposed detailed proceedures to be followed during coding and integration times.

3.3.5 Interface control

3.3.6 Subcontractor / vendor control

The configuration leader will track all upgrades and bug reports of the GitHub, TortoiseGit, and NetBeans IDE tools. A backup plan will be developed by the CM leader that will be utilized in the event of GitHub and TortoiseGit failing, becoming discontinued, or no longer supported. This plan will be submitted to the Project Manager within 2 weeks of the release of this document (Version 1.0.0). A copy of this plan is available in Annex A.

3.3.7 Release management and delivery

Release of versions must be approved by the project manager.

3.4 SCM schedules

The schedule for configuration management reporting, archiving, and upgrading is shown below in Figure 1.

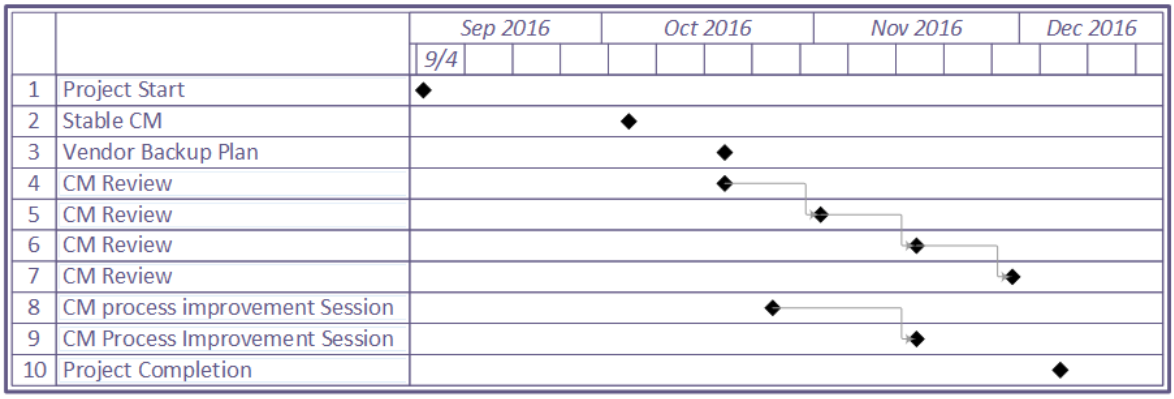

Figure 1: Confuguration Management Schedule

3.5 SCM resources

The configuration leader will require an estimated average of 2 hours a week to maintain the system configuration for the duration of the project. Time spent on CM activities by other members of the team are not accounted for here.

3.6 SCM plan maintenance

Due to the importance of a stable SCM plan, all changes to this document must be approved by the configuration leader and the project manager.

The configuration leader will do the following for the CM process improvement sessions:

- Review the effectiveness of this plan
- Quantify losses due to defects in this plan
- Review the effectiveness of GitHub and TortoiseGit
- Investigate the literature for new CM methods, quantify the costs and benefits of improvements
- Investigate new CM tools
- Suggest specific improvements to this CM process
- List the benefits of improvements

AnnexA: Configuration Management Tool Backup Plan

The Configuration Leader will obtain and dedicate a USB flash drive for this project. Once a week the Configuratin Leader will create a new clone of the current configuration of the project from GitHub on the USB drive. All previous clone versions will be retained. In the event of a failure or discontinunation of the GitHub service all requests to access files will be sent to the Configuration Leader. Any changes will then be returned to the Configuration Leader for integration.

Annex B: Bibliography

Braude, Eric J., and Michael E. Bernstein. *Software Engineering: Modern Approaches.* 2nd ed. Hoboken, NJ: J. Wiley & Sons, 2011. Print.