

# **Software Verification and Validation Plan**

## **IEEE 1012 - 2004**

Project: Java Air

Prepared by: Matthew D Moscatel and Xu Wu

**Team: Avian Limited**

## Contents

1. Purpose .....	3
2. Referenced documents.....	3
3. Definitions .....	3
4. V&V overview .....	4
4.1 Organization* .....	5
4.2 Master schedule .....	6
4.3 Software integrity level scheme .....	7
4.4 Resource summary .....	7
4.5 Responsibilities .....	7
4.6 Tools, techniques, and methodologies.....	8
5. V&V processes.....	8
5.1 Management of V&V .....	8
5.2 Acquisition V&V .....	9
5.3 Development V&V.....	9
5.4 Operation V&V .....	9
5.5 Maintenance V&V .....	9
6. V&V reporting requirements.....	9
6.1 Reporting .....	9
6.2 Administrative .....	9
6.3 Documentation .....	10
7. V&V administrative requirements .....	10
7.1 Anomaly reporting and resolution.....	10
7.2 Task iteration policy .....	10
7.3 Deviation policy .....	10
7.4 Standards, practices, and conventions.....	10
8. V&V documentation requirements.....	10

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Description</b>
1.0	10/5/2016	M. Moscatel	Document Creation
1.1	11/15/2016	Xu Wu	1. Added contents to section 2 2. Added contents to section 4.1 3. Revised section 4.2, 4.4 and 4.5 4. Added contents to section 6.1, 6.2 and 6.3 5. Added contents to section 7.2
1.2	12/4/2016	M. Moscatel	4.6 Added Github as a tool 7.3 Added deviation acceptance 7.4 Added standard reference 8. Added deviation acceptance

## 1. Purpose

This document describes the means by which the Java Air project will produce and maintain a high quality product. This maintenance will be performed through the verification and validation of the input/output expected management as well as the verification of source code in real time. The Java Air project is Java-based software that simulates the management software of an airline company. The scope of this document comprises the artifacts of all releases.

## 2. Referenced documents

- Software Engineering Modern Approaches, 2<sup>nd</sup> ed. By Eric J. Braude, Michael E. Bernstein.
- IEEE Standard 1012-2004, IEEE Standard for Software Verification & Validation Plan.

### 3. Definitions

CI = Configuration Item

CMMI = Capability Maturity Model Integration

IEEE = Institute of Electrical and Electronics Engineers

QA = Quality Assurance

SEI = Software Engineering Institute

SCMP = Software Configuration Management Plan

SPMP = Software Project Management Plan (this document)

SRS = Software Requirements Specification

SDD = Software Design Document

SQAP = Software Quality Assurance Plan

SVVP = Software Verification and Validation Plan

STP = Software Test Plan

UD = User Documentation

WBS = Work Breakdown Structure

U/PD = User/Product Director

PM = Project Manager

RE = Requirement Engineer

SA = Software Architect

IE = Integration Engineer

TE = Testing Engineer

CD = Code Developer

PNW = Purdue University Northwest

## 4. V&V overview

The V & V approach as described in IEEE 1012-2004 will be used for conducting project V & V activities. The activities will be planned and scheduled per the project schedule, the SRS, the PQP, and the availability of Cis.

### 4.1 Organization

The V&V process is tightly related to other phases of the software life cycle.

Specifically:

#### a. Project Management

The Project Manager (PM) is responsible for assigning human resources to take the Quality Assurance (QA) who are responsible for assuring the quality of the software project. When scheduling tasks, the PM shall take into consideration of the V&V work.

#### b. Requirement Analysis

The Requirement Engineers (RE) compose the Software Requirement Specification (SRS) and shall have the SRS reviewed by QA. Besides, the QA shall monitor or interview RE to extract information about how the requirement engineering process is carried out. The QA shall assure that the requirement engineering deliverables truly describes the customer's' needs.

#### c. Design

The Designers compose the Software Design Document (SDD) and shall have it reviewed by the QA. The QA shall assure that the design satisfies all the requirements defined in SRS.

#### d. Implementation

The Coders write code based off design. QA shall review code to assure the quality and that the code matches the design.

#### e. Testing

The Test Engineers (TE) builds and executes tests against code. QA shall review the tests to assure the quality and completeness of the tests and that the tests match code.

#### f. Deployment

QA shall review the deployment process to assure the quality of the process and verify that the delivered software is in working condition and meets all requirements defined in SRS.

g. Maintenance

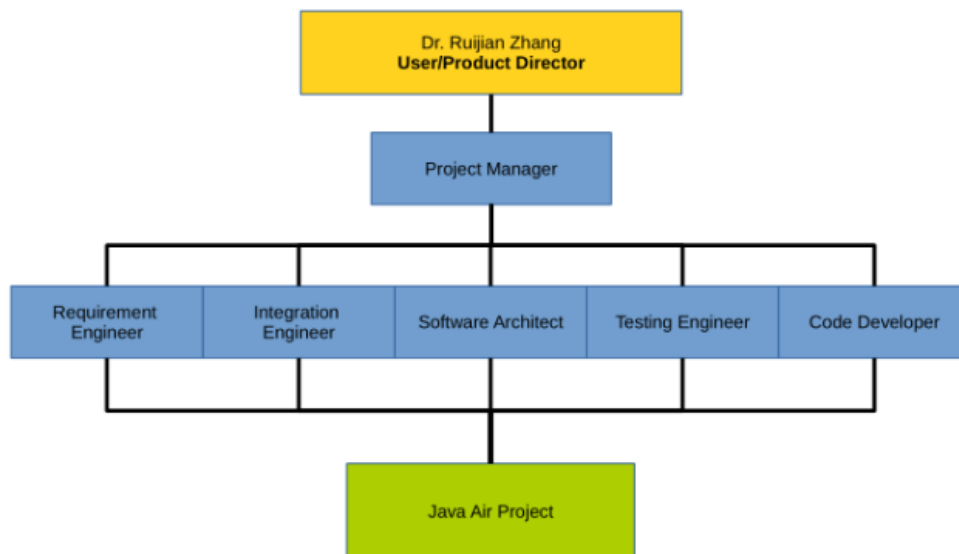
QA shall review all maintenance processes to ensure the quality and maintain consistency and stability of the software.

h. Configuration Management

QA shall review the configuration management plan to assure the quality of the process to support the software development activities.

The responsibilities of the participants in the project are show below:

<i><b>Members</b></i>	<i><b>Requirement Engineer</b></i>	<i><b>Integration Engineer</b></i>	<i><b>Project Manager</b></i>	<i><b>Testing Engineer</b></i>	<i><b>Software Engineer</b></i>
<i><b>Liaison Responsibility</b></i>	U/PD, PM	U/PD, PM	U/PD, All Members	PM	PM, RE
<i><b>Document Responsibility</b></i>	SRS	SCMP & UD	SPMP	SQAP, SVVP & STP	SDD



## 4.2 Master schedule

Because of the organization of V&V as described in Section 4.1, the V&V schedule follows directly to the project schedule as defined in the SPMP.

### **4.3 Software integrity level scheme**

IEEE 1012-2004, Section 4, provides guidance on selection of criticality levels for software based on its intended use and application. Criticality levels are established by a subjective evaluation of attributes. IEEE 1012 uses Integrity Levels to quantify criticality. The assigned Software Integrity Levels may vary as the software evolves. However, the software and hardware developed for client interaction related portions of this project will be used in a functional critical application and shall be classified as Software Integrity Level 4 (Criticality-High).

The project documents listed below identify the types of design outputs at the system level and will be assigned a Software Integrity Level 4 rating:

- 1) Project Plans, Software V&V Plan.
- 2) Project Specifications/Reports
  - a. Software Requirements Specification (SRS)
  - b. Software Design Description (SDD)
  - c. Validation Test Specification
  - d. Verification Test Specification
  - e. V&V Activity Summary Reports
- 3) Database Design Integration Drawings.
- 5) Verification and Validation Test Produces, Test Reports, Final V&V Report.

### **4.4 Resource summary**

The Java Air application is required to have low integrity.

### **4.5 Responsibilities**

As stated in organization, the responsibilities between Xu and Matt will

- Perform verification activities throughout the project, including inspections with in all phases.
- Perform and document all unit testing using JUnit
- Validate SRS, SDD, Code, and Test documentations together with other engineers.

- Verify that the team has followed its documented procedures, including those described in this document.
- Perform all post-unit testing
- Report the results to the team and management
- Maintain this document

#### 4.6 Tools, techniques, and methodologies

As it states in Resource summary, Java Air is required to run smoothly with low resource dependency. Stress testing will be conducted by measuring resource management while functionality is looped. We are using Microsoft Word for document creation, PDF writer for PDF conversion, Github for document control, and Google drive for document sharing.

### 5. V&V processes

The following explains the correlation of the Invensys Java air life cycle to IEEE 1012-2004 life cycle processes and activities.

IEEE 1012 V&V Lifecycle Process	Java Air Lifecycle Process
Management	Performed throughout lifecycle
Acquisition	Requirement Engineer Acquisition
Development	Development
*Concept	SRS document interpretation
*Requirements	SRS document interpretation
*Design	Architect Design
*Implementation	Integration
*Test	QA Testing
Operation	Deliver to Customer
Maintenance	Real time maintenance only, program won't be used after course is concluded

#### 5.1 Management of V&V

The Management process is applicable to all phases the Project. Java Air management shall meet the task performance requirements for management of V&V as stated in IEEE 1012-2004.



## **5.2 Acquisition V&V**

The Requirement engineers are responsible for acquiring all relevant information from the client. These requirements will be documented in the SRS.

## **5.3 Development V&V**

The Architect and Integration engineer will be responsible for developing and integrating the source code, GUI, and compiled code.

## **5.4 Operation V&V**

This phase covers the operation of the software product and operational support to users after installation normal commissioning. It addresses operational testing, system operations, and user support with respect to the operating procedures. It is the responsibility of the Testing engineers to test the operational integrity of the software.

## **5.5 Maintenance V&V**

We will not be performing Maintenance outside of real time when needed. This software is not meant for commercial use after the conclusion of the course.

# **6. V&V reporting requirements**

V&V reporting shall occur throughout the entire life cycle and include the following reporting mechanisms.

## **6.1 Reporting**

The QA engineers attached to the project directly report the status of V&V weekly to the Project Manager and the major professor.

Summary reports are required for the following phases:

- Requirements Phase
- Design Phase
- Implementation Phase
- Test Phase

## **6.2 Administrative**

The Project Manager is responsible for ensuring that all V&V reporting is performed.

## **6.3 Documentation**

A single report that includes all versions of the results of the tasks described in this document is maintained

## **7. V&V administrative requirements**

### **7.1 Anomaly reporting and resolution**

Anomaly reporting will be defined in PPM 10.0, Nonconformance and Corrective Action.

### **7.2 Task iteration policy**

All task interactions will be logged with correct nomenclature.

V&V tasks will be repeated at the discretion of the QA engineers, but these will include the following criteria:

- An inspection whose defects count is more than 20 percent greater than the norm
- A test whose defects count is more than 20 percent greater than the norm
- An entire phase if the previous phase changes by more than 20 percent

### **7.3 Deviation policy**

Deviations from any standard will be corrected by the responsible party. Deviations outside of scope or deemed inefficient due to time constraints will be signed off by the project manager. Deviation considered a benefit will need to be signed off by the project manager and the parties responsible.

### **7.4 Standards, practices, and conventions**

[Software Configuration Management IEEE 828](#)

[Software Test Documentation IEEE 829](#)

[Software Requirements Specification IEEE 830](#)

[Software Verification and Validation IEEE 1012](#)

[Software Design Description IEEE 1016](#)

[Software Project Management IEEE 1058](#)

[Software User Documentation IEEE 1063](#)

## 8. V&V documentation requirements

The Verification process will include documenting what code and documentations have been reviewed and on which date. Each nonconformance and anomaly will be documented as corrective actions and the closing of said corrective action would be documented for completion. Each nonconformance will be given a unique nonconformance number for tracking purposes. Nonconformance numbers will be classified based on which part of the development stage it was found, what are the effects of the correction, and responsible party. Non-conformances and deviations that are either outside the scope of the allotted time or are considered a benefit to the functionality of the project must be accepted by the project manager.

Validation testing will consist of input and output QA testing. Every test performed will be given a unique test number. If a test fails, said test will be associated with a unique nonconformance number.