Steve Jia

ssjia@pnw.edu

Project: Java Air

Team: Avian Limited  
Project Manager: Steve Jia  
Advanced Software Engineering Fall Semester 2016  
Dr. Ruijian Zhang

Software Project management plan

IEEE 1058

Contents

[**Approvals** 2](#_Toc462929209)

[**Revision History** 2](#_Toc462929210)

[**1.** **Overview** 3](#_Toc462929211)

[1.1. Project Summary 3](#_Toc462929212)

[1.2. Evolution of the SPMP 3](#_Toc462929213)

[2. References 3](#_Toc462929214)

[3. Definitions 3](#_Toc462929215)

[4. Project Organization 4](#_Toc462929216)

[4.1. External Interfaces 4](#_Toc462929217)

[4.2. Internal Structure 4](#_Toc462929218)

[4.3. Roles and Responsibilities 4](#_Toc462929219)

[5. Managerial Process Plan 6](#_Toc462929220)

[5.1. Project Start-Up Plan 6](#_Toc462929221)

[5.1.1. Estimation Plan 6](#_Toc462929222)

[5.1.2. Staffing Plan 6](#_Toc462929223)

[5.1.3. Resource Acquisition Plan 6](#_Toc462929224)

[5.1.4. Project Staff Training Plan 6](#_Toc462929225)

[5.2. Work Plan 8](#_Toc462929226)

[5.2.1. Work Activities 8](#_Toc462929227)

[5.2.2. Schedule Allocation 8](#_Toc462929228)

[5.2.3. Resource Allocation 8](#_Toc462929229)

[5.2.4. Budget Allocation 8](#_Toc462929230)

[5.3. Control Plan 8](#_Toc462929231)

[5.3.1. Requirements Control Plan 9](#_Toc462929232)

[5.3.2. Schedule Control Plan 9](#_Toc462929233)

[5.3.3. Budget Control Plan 9](#_Toc462929234)

[5.3.4. Quality Control Plan 9](#_Toc462929235)

[5.3.5. Reporting Plan 9](#_Toc462929236)

[5.3.6. Metrics Collection Plan 9](#_Toc462929237)

[5.4. Risk Management Plan 9](#_Toc462929238)

[5.5. Project Closeout Plan 10](#_Toc462929239)

[6. Technical Process Plan 10](#_Toc462929240)

[6.1. Process Model 10](#_Toc462929241)

[6.2. Methods, Tools, and Techniques 10](#_Toc462929242)

[6.3. Infrastructure Plan 11](#_Toc462929243)

[6.4. Product Acceptance Plan 11](#_Toc462929244)

[7. Supporting Process Plan 11](#_Toc462929245)

[7.1. Configuration Management Plan 11](#_Toc462929246)

[7.2. Verification and Validation Plan 11](#_Toc462929247)

[7.3. Documentation Plan 11](#_Toc462929248)

[7.4. Quality Assurance Plan 11](#_Toc462929249)

[7.5. Testing Plan 11](#_Toc462929250)

[7.6. Requirements and Specification Plan 11](#_Toc462929251)

[7.7. Design Plan 12](#_Toc462929252)

[8. Additional Plans 12](#_Toc462929253)

[8.1. Communication Plan 12](#_Toc462929254)

# **Approvals**

|  |  |  |
| --- | --- | --- |
| **Title** | **Signature** | **Date** |
| Project Manager / Author |  |  |
|  |  |  |
|  |  |  |

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 2016-09-20 | S. Jia | Document Creation |
| 1.0.1 | 2016-09-21 | S. Jia | Added index items for 5.1, 5.2, and 5.3;  Added main contents for sections 1, 2, 3, and 4. |
| 1.0.2 | 2016-09-29 | S. Jia | Finished section 5;  Finished section 6;  Added content for section 7; |

1. **Overview**
   1. Project Summary

This project is organized to produce a software application for Java Air’s customers and employees. The overall plan for this product is planning, design, implementation, and testing. This product will be developed in several stages, with each stage following review and demonstration of each version. The early versions of the product will mainly be basic graphical user interfaces and animations, and with advances in development, later versions of the product will have database support and more advanced graphical user interfaces and animations, as well as error checking and logging functionalities.

* 1. Evolution of the SPMP

This document will be maintained on a weekly basis by the project manager and leader, Steve Jia. Contents of this document will be improved and maintained based on the development progress of the project team. It is subject to configuration management by means of the Software Configuration Management Plan (SCMP). It is the project manager’s responsibility to submit this document as a Configuration Item (CI), and to keep it up to date. This Software Project Management Plan (SPMP) mainly follows the format of IEEE 1058.1-1998.

1. References

[IEEE] The applicable IEEE standards are published in “IEEE Standards Collection,” 1997 Edition.

[Braude] The principle source of textbook reference material is *Software Engineering: An Object-Oriented Perspective* by E. Braude (Wiley, 2000).

1. Definitions

**CI** = Configuration Item

**CMMI** = Capability Maturity Model Integration

**IEEE** = Institute of Electrical and Electronics Engineers

**QA** = Quality Assurnace

**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

1. Project Organization
   1. External Interfaces

The project team will interface with the following individuals and organizations: Dr. Ruijian Zhang (U/PD), for technical and standards direction, as well as requirements and specifications. The project team may also interface with members of PNW for tips and ideas.

* 1. Internal Structure

Figure SPMP.1 shows the organization of the Java Air project within Avian Limited.

The project will be organized as a team of peers with designated roles. The roles are project manager, requirement engineer, software architect, integration engineer, testing engineer, and code developer. Each project team member has another role besides the code developer role.

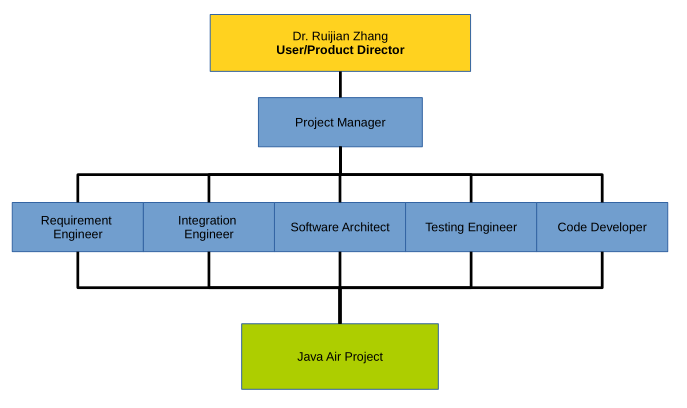


Figure SPMP. : Organization of Avian Limited

* 1. Roles and Responsibilities

The responsibilities of the participants in the project are show in Table SPMP.1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Members** | Requirement Engineer | Integration Engineer | Project Manager | Testing Engineer | Software Engineer |  |  |
| Liaison Responsibility | U/PD, PM | U/PD, PM | U/PD, All Members | PM | PM, RE |  |  |
| Document Responsibility | SRS | SCMP & UD | SPMP | SQAP, SVVP & STP | SDD |  |  |

Table SPMP. : Avian Limited Members, Roles, and Responsibilities

Each role has its own set of responsibilities.

* The project manager will be responsible for the “health” and progress of the project as well as of each team member. The project manager will also be responsible for creating and maintaining the scope of the project, the schedule of the project, and task assignments of the project. The project manager is also responsible for creating and maintaining the Software Project Management Plan.
* The requirement engineers will be responsible for establishing the requirements and specifications for the software product. The requirement engineers will meet with team members to discuss requirements and then review them with the U/PD. The requirement engineers are also responsible for creating and maintaining the Software Requirements Specification. Because the SRS will be completed early on during the project, the requirement engineers will be asked to assist other team members with their documentation, especially that of the testing engineers’.
* The software architect will be working closely with the requirement engineers to translate the SRS into functional specifications and UML diagrams. The software architect will also list different software classes and units with their respective inputs and outputs. The software architect is also responsible for creating and maintaining the Software Design Document.
* The integration engineer is primarily responsible for designing the user interfaces, integrate the interfaces with the code-base, and performing product demonstrations to the team and to the U/PD. The integration engineer is also responsible for creating and maintaining the Software Configuration Management Plan as well as the Software User Documentation.
* The testing engineers are primarily responsible for reviewing and testing of the software product. The testing engineers will establish test plans and record test results. They will also report debugs and issues to the code developers. The requirement engineers are also responsible for creating and maintaining the Software Quality Assurance Plan, the Software Verification and Validation Plan, and the Software Testing Plan. Due to Matt (TE)’s expertise in database, he will be designing and implementing the database for the product. Matt will also take primary responsibility for creating SQL queries.
* The code developers will be responsible for creating the code base and implementing the software interfaces that were designed. They will also perform functioning point analysis on each unit, and also perform bugs and issues fixes on the code-base.

Being Responsible for a document includes the following:

* Making sure that the document is created on time
* Having the project manager identify the writer(s) of the document
* Keeping the document up-to-date throughout the project life cycle

1. Managerial Process Plan
   1. Project Start-Up Plan
      1. Estimation Plan

Due to inexperience of most of the team members, it’s hard to estimate this project’s cost based on previous projects or experiences. Therefore, estimation of this project’s duration and cost will be made first based on the Work Breakdown Structure, the Network Diagram, and will be updated when the high-level requirements and the detailed specifications and designs are established.

The work breakdown structure (Figure SPMP.2; next page) was created based on a team activity lead by the project manager. During this activity, team members worked together first to identify project deliverables, then establish steps and items that each team member will do in order to complete this project. All the work packages (tasks) are then compiled together to create a network diagram (Figure SPMP.3) to illustrate the estimated duration and cost of the project.

* + 1. Staffing Plan

The roles will be filled as follows.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Members** | CAO, Yuwei | EXRLEBEN, Amy | JIA, Steve | MOSCATEL, Matt | QI, Guoyu | WU, Xu | ZHANG, Rui |  |  |
| Role | Requirement Engineer, Code Developer | Integration Engineer, Code Developer | Project Manager, Code Developer | Testing Engineer, Code Developer | Requirement Engineer, Code Developer | Testing Engineer, Code Developer | Software Architect, Code Developer |  |  |

Table SPMP. : Avian Limited Roles Assignment

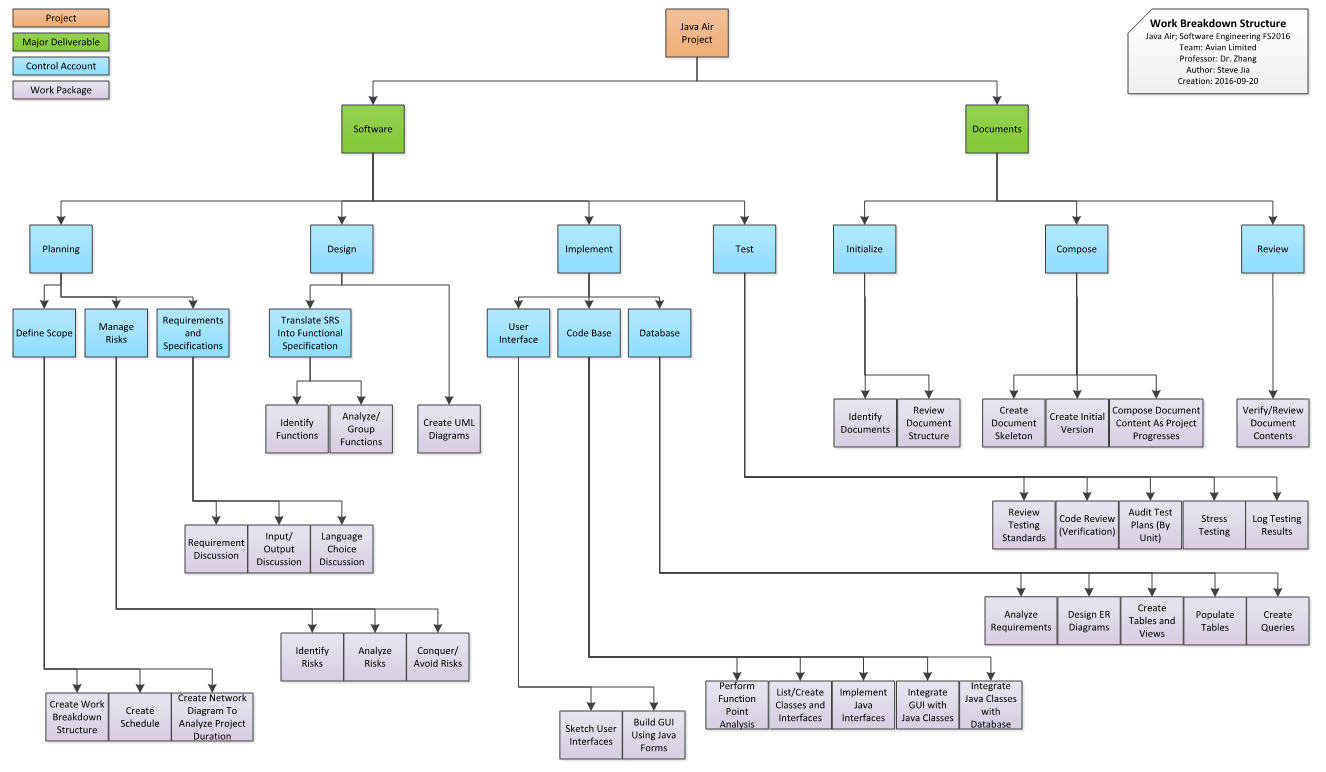
* + 1. Resource Acquisition Plan

Each team member will use his/her person computer as well as university (PNW) computers to work on the project. Software tools used for this project will be primarily free-ware, but each team member can use their personally-licensed software if desired.

* + 1. Project Staff Training Plan

All team members whose are not proficient in Java should study the PowerPoint slides from the Object-Oriented Programming Design course (Purdue University Calumet, Fall Semester 2015) that are located on the shared Google Drive folder. Team members should also refer to the ***Oracle Java Online Tutorials*** (<https://docs.oracle.com/javase/tutorial/>) for guidance and official documentations. Team members should also increase their Java programming skills by working on practice questions that are provided by ***CodingBat*** (<http://codingbat.com/java>).

Figure SPMP. : Java Air Project Work Breakdown Structure



* 1. Work Plan
     1. Work Activities

The work on this project will be divided into configuration management, quality assurance (including testing), requirements analysis, design, and implementation. The project roles and responsibilities are show in Table SPMP.1 as well as a detailed list on page 5.

* + 1. Schedule Allocation

The schedule is shown in Figure. Please refer to the Software Quality Assurance Plan (SQAP) for the schedule of quality activities. <***insert high-level schedule here***>

* + 1. Resource Allocation

The work breakdown structure in Figure## shows the bottom line of person-months available each month.

Currently design has not completed, so task assignment is not available. Names will be added once design and configurations have been determined.

* + 1. Budget Allocation

The Java Air project does not have a dollar-budget, therefore the budget for this project has been determined by the Project Manager to allow an average of 4 hours per engineer per week. Therefore, in a typical week, 28 hours are budgeted for the project. The monthly engineer-hour budget takes in consideration for the Labor Day National Holiday, PNW Fall Break and Thanksgiving Break.

Table SPMP. : Java Air Budget Allocation

|  |  |
| --- | --- |
| **Month Number** | **Allocation (hours)** |
| 1 - September | 105 |
| 2 – October | 112 |
| 3 – November | 101.5 |
| 4 – December | 28 |
| **Total** | 346.5 |

* 1. Control Plan

In the initiation and planning phase of the Java Air project, the entire team will meet every Saturday evening from 5:00 p.m. to 6:00 p. m. and whenever a meeting is needed. Unscheduled meetings can take place after class, or any time team members are available. The project manager will schedule extra meetings with the team members after each class on Mondays and Wednesdays.

During the requirement analysis and design phases of the project, the requirement engineers, the software architect, and the project manager will meet Tuesday afternoons to review specifications and design documents.

For the First week of October when the Project Manager is unavailable, Testing Engineer Matt will be the substitute Project Manager. The substitute PM will coordinate meetings with the rest of the team members and communicate meetings to the Project Manager.

* + 1. Requirements Control Plan

The requirement engineers (Cao, Qi) will report to the project leader on the status of the Software Requirements Specification in-person or in writing each Monday.

* + 1. Schedule Control Plan

The project manager (Jia) will report to the team on the status of the schedule in-person and/or in writing each Wednesday or Saturday.

* + 1. Budget Control Plan

The project leader will report to the U/PD on the status of the budget in writing in each meeting minutes.

* + 1. Quality Control Plan

The testing engineers (Moscatel, Wu) will provide written reports to the project manager and carbon copy rest of the team members.

* + 1. Reporting Plan

The written reports referred to in this section (5.3) will be via e-mail.

* + 1. Metrics Collection Plan

Please see section ## <***insert section here***> of the Software Quality Assurance Plan.

* 1. Risk Management Plan

Figure ## <insert figure> shows a format for risk reporting and retirement. Starting from the design phase, each team meeting will have an agenda item for risk identification brainstorming and reporting on risks that have been identified.

* + 1. Risk #1: Java Expertise

Based on the language-choice decision table, Java is the most suitable language choice for the Avian Limited team members for the Java Air project. However, four out of seven team members identified themselves as having little or no knowledge at all about Java and object-oriented programming.

Using Java to implement the Java Air code base is crucial, and most of the team members are fluent in Java is a critical issue that has very high impact. By interviewing the team members that lack Java programming experiences, the project manager found that they lack programming experience in general.

Because of this, risk retirement by avoidance is not a viable option, since these team members do not know how to code in any other major programming languages. Therefore, this risk must be conquered. The project manager will provide guidance materials on the Java programming language, reference to the official Oracle Java documentations, and URL to a Java coding-exercise website.

* + 1. Risk #2: HTML Expertise

Originally, the team thought having a website as the graphical user interface, however, after interview, all the team members are identified as “little or no experience in HTML and Javascript”. This issue can cause project delays and software performance issues later on.

After communicating with the U/PD and the team members, the project manager decided to retire this risk by avoidance. The team will use Java JFrames/Swing Forms, and an IDE like Netbeans, to create the graphical user interface.

* + 1. Risk #3: Database Integration Issues

After preliminary discussion, the team decided to use a database to store the project’s data and act as the back-end for the user interface and Java classes. However, most of the databases require a server, which was not achievable.

The back-end is very important to the project. The database can store all the project information and software configurations, and it provides easy access to data to the user interface. Also, one of the most important things is to have this database offline, running locally with the rest of the software product using SQL queries to communicate.

The project manager researched possible products and found SQLite, which should satisfy all the requirements. The team retired this risk by conquest.

* 1. Project Closeout Plan

The Java Air software product will not be maintained beyond December, 2016. At the end of the project, when the software testing is completed and all the deliverables are reviewed, the project manager will deliver the items to the U/PD and also submit a final project report and a team performance and participation document.

1. Technical Process Plan
   1. Process Model

The Java Air project will be executed using a Unified process development process. <*explain details of how unified process used*>

* 1. Methods, Tools, and Techniques

The Java Air project will use UML for design and will be implemented in Java.

For planning and documentation, Microsoft Word, Microsoft Excel, and Microsoft Visio will be used. If team members need free versions of similar software, they can use LibreOffice products.

For design, StarUML will be used to sketch diagrams and document classes, reference variables, methods, relationships, and hierarchy.

For implementation, Java will be the language of choice. Team members can use IDEs such as Netbeans and Eclipse for development.

Github.com, git, and tortoiseGit will be used for version control.

* 1. Infrastructure Plan

The Java Air software product will require­ a computer with at least a single CPU core and at least 2 GB of memory. The computer must be running a Windows 7 or later operating system. The computer might also need Java installed. The computer should also have a mouse and keyboard in order to operate the software product.

* 1. Product Acceptance Plan

The U/PD and the project manager will finalize acceptance criteria prior to the last six weeks of the project’s end. Acceptance tests will be determined later in the project as implementation and testing are completed.

1. Supporting Process Plan

*Notes to the integration engineer: the project manager created configuration-item numbers for the documents. The CI number will be JAD (Java Air Documents) followed by a dash, and then by a single digit from 1 to 8.*

* 1. Configuration Management Plan

The Java Air Configuration Management Plan, configuration item ***JAD-2***, will be created and maintained by the integration engineer.

* 1. Verification and Validation Plan

The Java Air Software Verification and Validation Plan, configuration item ***JAD-6***, will be created and maintained by the testing engineers.

* 1. Documentation Plan

The Java Air Software User Documentation Plan, configuration item ***JAD-8***, will be created and maintained by the integration engineer.

* 1. Quality Assurance Plan

The Java Air Software Quality Assurance Plan, configuration item ***JAD-5***, will be created and maintained by the testing engineers.

* 1. Testing Plan

The Java Air Software Testing Plan, configuration item ***JAD-7***, will be created and maintained by the testing engineers.

* 1. Requirements and Specification Plan

The Java Air Software Requirements and Specification, configuration item ***JAD-3***, will be created and maintained by the requirement engineers.

* 1. Design Plan

The Java Air Software Design Plan, configuration item ***JAD-4***, will be created and maintained by the software architect.

1. Additional Plans
   1. Communication Plan