FEED ME

Project Management Plan

Version 1.0

Document Number: SPMP-001

**Project Team: **A23****

**Project Members: **Peter Holoubek** (ph1335), **Teddy Kim** (tk1931), **JinZhao Su** (js9202)**

# 

# REVIEW AND APPROVALS

|  |  |  |  |
| --- | --- | --- | --- |
| Printed Name | Function | Date | Signature |
| Theodore Kim | Author | April 9, 2019 |  |
| JinZhao Su | Author | April 9, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/52861846_2336825663028189_6860368647928414208_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=8bb7d8bce3fedf38668fe3f9d05d7a6b&oe=5D258DEB |
| Petr Holoubeck | Author | April 9, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/s2048x2048/52825818_155163132053922_2270601179052376064_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=4b9107141ccddee8f9ebf13e5428e56c&oe=5CDB1CAF |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| Date | Revision Number | Purpose |
| April 9, 2019 | Version 1.0 | Initial Document Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

TABLE OF COTENTS

[REVIEW AND APPROVALS 2](#_Toc6309257)

[REVISION HISTORY 3](#_Toc6309258)

[1. OVERVIEW 1](#_Toc6309259)

[1.1 Project Summary 1](#_Toc6309260)

[1.2 Purpose, Scope and Objectives 1](#_Toc6309261)

[1.3 Assumptions and Constraints 2](#_Toc6309262)

[1.4 Project Deliverables 2](#_Toc6309263)

[1.5 Schedule and Budget Summary 3](#_Toc6309264)

[1.6 Evolution of the Plan 4](#_Toc6309265)

[2. REFERENCES 4](#_Toc6309266)

[3. DEFINITIONS 4](#_Toc6309267)

[3.1 Project Definitions 4](#_Toc6309268)

[3.2 Project Abbreviations 4](#_Toc6309269)

[4. PROJECT ORGANIZATION 5](#_Toc6309270)

[4.1 External Interfaces 5](#_Toc6309271)

[4.2 Internal Structure 5](#_Toc6309272)

[4.3 Roles and Responsibilities 5](#_Toc6309273)

[5. MANAGEMENT PROCESSES 6](#_Toc6309274)

[5.1 Start-Up Plan 6](#_Toc6309275)

[5.2 Work Plan 6](#_Toc6309276)

[5.3 Control Plan 8](#_Toc6309277)

[5.4 Risk Management Plan 9](#_Toc6309278)

[5.5 Post Implementation Plan 9](#_Toc6309279)

[6 TECHNICAL PROCESSES 10](#_Toc6309280)

[6.1 Process Model 10](#_Toc6309281)

[6.2 Methods, Tools and Techniques 10](#_Toc6309282)

[6.3 Infrastructure Plan 11](#_Toc6309283)

[6.4 Product Acceptance and Migration Plan 11](#_Toc6309284)

[7 SUPPORTING PROCESSES PLAN 11](#_Toc6309285)

[7.1 Configuration Management Plan 11](#_Toc6309286)

[7.2 Qualification (Verification and Validation) Plan 12](#_Toc6309287)

[7.3 Documentation (Library) Plan 12](#_Toc6309288)

[7.4 Quality Assurance Plan 12](#_Toc6309289)

[7.5 Reviews and Audits 13](#_Toc6309290)

[7.6 Problem Resolution Plan 14](#_Toc6309291)

[7.7 Environmental Management Plan 14](#_Toc6309292)

[7.8 Process Improvement Plan 14](#_Toc6309293)

[8 ADDITIONAL PLANS 14](#_Toc6309294)

[9 INDEX 14](#_Toc6309295)

[10 RATIONALE 14](#_Toc6309296)

[11 NOTES 15](#_Toc6309297)

[12 APPENDICES 16](#_Toc6309298)

[12.1 Schedule Tracking 16](#_Toc6309299)

[12.2 Defect Tracking 17](#_Toc6309300)

[12.3 Project Schedule Gantt Chart 18](#_Toc6309301)

## 1. OVERVIEW

### Project Summary

The motivation for the project is that online food recommendation services require some amount of vetting and research as services such as Yelp may have ingenuine reviews and ratings. Therefore, a service that performed quality control and screening is a need among said consumers. Hence, the customer’s expectations are that the product *makes a decision* for the user rather than just offer a search functionality through a database of restaurant options. This decision should require minimal user input and should be representative of their individual preferences.

The product can be used not only in the daily lives of individuals looking to eat out, but also o travelers looking to become familiar with local cuisines and restaurants looking to advertise their business to potential customers.

This document’s purpose is to provide guidance for the project team on the management, organization and development of the project during later lifecycle stages. This document includes plans for the management, technical and supporting processes.

While the primary audience of the SRS was the project customer, the SPMP is targeted at the developers and managers of the project. It is intended to provide guidance for the project team in the development of the product and the progression of the project. The schedules and budgets provided in this document should direct the actions of the project team.

### Purpose, Scope and Objectives

The proposed system includes a means of making decisions for users based upon their individual preferences (which they provide to the product) and restaurant data provided by 3rd party application programming interfaces (APIs). It also provides a means for businesses to advertise to the product’s users and target audiences that have demonstrated behaviors and preferences compatible to said businesses. It is important to distinguish that the system DOES NOT track information about individual restaurants (location, food type, hours of operation, etc.) but rather receives that information from the use of APIs provided by other services which do track that information. The system simply aggregates said data and matches individual’s preferences to said restaurant choices.

The primary objective of the system is to provide a food choice as good as if an individual made the choice themselves. This project is will be delivered as a single, completed product at the designated delivery date.

This project is not critical; however, it should be treated with high priority because competitor research has revealed several other projects currently under development that are seeking to accomplish the same goals as the proposed product, therefore, development of the project should not be delayed significantly in order to gain some advantage in the potential market.

Finally, the primary business need to be satisfied by the release of the product will be to produce a product which will allow businesses to take advantage of the plethora of data on the behavior of consumers as well as to provide consumers the ability to take advantage of the restaurant data available from internet consumer services.

### Assumptions and Constraints

The project requirements assume that datasets for 3rd party food review services are both public and allowed to be commercialized by other businesses and services. Upon investigating specific services, if said datasets are not able to be referenced, the requirements for the project would change significantly as it affects the core functionality of the proposed product.

Another assumption made by the specification is that food preference can be predicted using a computer, mathematic model. While research has been done into the ability to make qualitative predictions based upon an input set using a computational model has been promising, it has been demonstrated that some relationships are too complex to be represented by a static model (i.e. a regression or a deep neural network). Therefore, if the decision relationship was unpredictable, it would greatly increase the feasibility of the project as well as its specified requirements.

The primary constraint for the project relates to the involvement of third-party services to provide the data for the product. Specifically, the project developer must adhere to the specific policies and terms of use of said services such that using said data from these services does not place the project in legal responsibility for potential misuse. Further details on these specific constraints will become further available as the project matures and specific APIs are chosen for use in the product.

As the product will be collecting personal information on individual users, developers must adhere to data protection laws in the municipalities in which the product will be released. For example, the Calif. Bus. & Prof. Code §§ 22575-22578 requires commercial websites to post a privacy policy specifically outlining how the site uses its user’s information.

Finally, the project will be constrained by the deadline implemented by the primary project stakeholder, Professor Fred Strauss and the NYU Tandon School of Engineering Computer Science and Engineering Department, which provides a hardlines and budget for the project (further clarified in future releases of this document and future documents).

### 1.4 Project Deliverables

The project deliverables are defined by the project authority. The deliverables for the documentation of the project are listed in Table 1. Aside from the project documentation, additional deadlines related to the development of the product are listed in the project schedule in later sections.

TABLE 1

Project Document Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable | Description | Deadline |
| Project Proposal | An outline of the project’s motivation, description of its requirements and constraints, description of its concept, and list of deliverables | February 16, 2019 |
| Software Business Specification | An outline of the business needs of the project, as well as a development of the product requirements and proposed decomposition of the product. | February 28, 2019 |
| Software Requirements Specification | Specification of the final requirements of the product, traceability to customer expectations and constraints, and verification plans for said requirements | March 28, 2019 |
| Software Project Management Plan | Definition of the project organization and proposed management of the project including planning for workflow, risk management, and technical management. | April 9. 2019 |
| Software Analysis Specification | A breakdown of the proposed product and its requirements decomposed into smaller functional units. A technical specification for proposed behavior and operation. Also a definition of acceptable software quality and human / environment interaction standards. | April 25, 2019 |
| Software Design Document | Final document for scope of this project | May 9, 2019 |

### 1.5 Schedule and Budget Summary

The project schedule is divided into various milestones imposed by the project authority for the delivery of the various documents. The project schedule includes the following milestones, processes, and supporting processes:

* **Business Definition Planning**
* **Milestone (submission for review):** SRS **-** Business Definition
* **Requirements Specification**
* **Milestone (submission for review):** SRS - Requirements
* **Project Management Planning**
* **Milestone (submission for review):** SPMP
* **Requirements and Project Analysis (High level design)**
* **Milestone (submission for review):** RAS
* **Detailed Design**
* **Milestone (submission for review)**: Software Design Document

See the appendix for an elaboration of the project schedule in the form of a Gantt Chart.

Budget management processes are not included in Version 1 of this document.

### 1.6 Evolution of the Plan

Following the initial development of the SPMP, the document will be reviewed for quality assurance via a desk check, peer review and a walkthrough. Then it will be submitted to the project authority for an inspection. Upon receiving the feedback from the project authority, the project team will correct the identified defects and release a revised version 1 (version 1.1). It will then be considered baselined and enter configuration and documentation management. Future changes to the document will be reviewed as a part of the management plans defined later in this document.

Possible changes to be made to the document include changes to the project schedule or budget as a result of slips or over spending. Management plans such as for risk and configuration management could be made if the processes planned in this document are found to be ineffective or faulty and need to be revised in order to improve said processes throughout the duration of the project.

## 2. REFERENCES

Project Proposal for Feed Me, Version 1 (February 10, 2019): Theodore Kim, JinZhao Su, Petr Holoubeck.

System Requirements Specification for Feed Me, Version 2 (April 2, 2019): Theodore Kim, JinZhao Su, Petr Holoubeck.

## 3. DEFINITIONS

### 3.1 Project Definitions

**Project Team:** the team of students

**Project Authority:** Professor Fred Strauss and the Software Engineering course standards

### 3.2 Project Abbreviations

**SPMP:** Software Project Management Plan

**SRS:** System Requirements Specification

**SQA**: Software Quality Authority \

**RAS:** Requirements and Analysis Specification

## 4. PROJECT ORGANIZATION

### 4.1 External Interfaces

The project team is simply responsible for the development of the Feed Me application. This application takes user input and 3rd party API restaurant and location data and generates useful decisions for use by the customer in located places to eat. Therefore, the project will need to interface with those external APIs (in the form of documentation) in order to produce the final product.

Furthermore, the project team will need to interface with external entities such as the project authority (who will be evaluating the development of the project), external vendors to purchase 2rd party, commercial products and services.

### 4.2 Internal Structure

There are three members of the project team. The project manager is Theodore Kim

### 4.3 Roles and Responsibilities

TABLE 2

Work Activities, Supporting Processes, and Responsible Parties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Work Activity | Supporting Processes | Activity Input | Activity Output | Responsible Party |
| Project planning | Brainstorming | Project Team Selection | Project idea  Project proposal | Entire Team |
| Business Definition | Customer expectation identification  Business Requirements Specification  Quality Control  Risk Management | Project proposal | Business Definition (SRS, Version 1) | Development Team, Customer, Project Authority, SQA |
| Requirements specification | Requirements specification  Use case specification  Quality Control | Project proposal  Business Definition | Requirements Specification (SRS, Version 2) | Development Team, Customer Project Authority, SQA |
| Project Management Planning | Scheduling  Work Activity Identification  Risk Management | Completed SRS | Project Management Plan (SPMP, Version 1) | Development Team, Customer Project Authority, SQA |
|  | Configuration Management  Quality Control |  |  |  |
| Requirements and Analysis (High Level Design) | Software Architecture Decomposition  Risk Analysis  Quality Control  Configuration Management  Verification and Testing | SPMP, SRS | RAS | Development Team, Customer Project Authority, SQA |
| Detailed Designing | Designing | SPMP, RAS, SRS | Design Document | Development Team, Customer Project Authority, SQA |

## 5. MANAGEMENT PROCESSES

### 5.1 Start-Up Plan

#### 5.1.1 Estimation Plan

The project team will utilize accomplish project estimation using analogous estimation. In other words, time requirements will be determined based upon what the previous project’s estimated completion time was compared to its actual time for completion. Then, each person’s estimate is averaged and then adjusted by the mean difference between the actual and predicted time requirements for the previous documents.

#### 5.1.2 Staffing Plan

We have all the people on the team to be able to complete the project sufficiently.

#### 5.1.3 Resource Acquisition Plan

Not included in Version 1of this document.

#### 5.1.4 Training Plan

As of version 1 of this document, there are no skills that have been identified as necessary for the development of the project which the project team does not possess. However, if, during development, the project team identifies technologies for which training is necessary, the project team will aim to train on these skills early. This requires that said skills are identified early on in the project lifecycle such that sufficient time is allocated for training using online resources.

### 5.2 Work Plan

#### 5.2.1 Work Activities

The work activities performed for the project are defined prior to the project starts. It is important to specify said work activities for both resource, risk and schedule management. They are:

* Project Planning: This planning stage is the inception of the project idea and involves the initial identification of the customer and their needs for potential problems.
* Business Definition: The definition and identification of the business requirements and goals for the project, the scope of the project and future system, and potential risks to the project.
* Requirements Specification: The identification and elaboration of requirements for the project and potential product.
* Project Management Planning: Making plans for the processes to be completed during the project, particularly, setting up plans for how each work activity will be completed, the plans for each plan, specifying how risks are managed and how generated artifacts are handled within the configuration database.
* Requirements and Analysis (High Level Design): A further refinement of the proposed product and project beyond the requirements specified in the SRS. This includes the creation of a software architecture.
* Detailed Design: The creation of a detailed design which will provide developers with a specific and well-thought-out guide for implementation.

Aside from the high level work activities, various supporting processes are completed throughout the project lifecycle which are important to the successful management of the project:

* Risk Management
* Configuration Management
* Schedule Management
* Resource Management
* Quality Assurances
* Testing and Verification

#### 5.2.2 Schedule Allocation

See Gantt Chart in the Appendix of this document for the full project schedule. The Gantt chart includes each deliverable required for the project and how each individual will be assigned to each work activity. Scheduling is based upon several factors. Mainly it is dependent on the deadline imposed for the project milestones by the project authority. Additionally, the initial project schedule was based upon speculation and the schedules of the individuals involved in the project. Of course, these estimations will most likely be inaccurate, therefore the schedule will be periodically revisited to ensure that the schedule is accurate and realistic. Extra time was paid attention to quality assurance during the initial schedule allocation.

#### 5.2.3 Resource Allocation

All activities will utilize the full (human) resources of the team. As of this stage in the development of the project, there are no other resources required for the project deliverables aside from hours of work committed by the project team. Therefore, being able to commit the full resources of the entire team to each work activity is achievable.

#### 5.2.4 Budget Allocation

Not included as of Version 1 of this document.

### 5.3 Control Plan

#### 5.3.1 Requirement Control and Traceability

The requirements specified in the SRS are derived from the expectations and preliminary requirements outlined in the Project Proposal document.

The requirements are each numbered with a unique identifier such that, later developments of the project baseline, can reference specific requirements. Using these and future identifiers, traceability can be maintained in future phases and iterations of the project. This traceability ensures that when a defect is found, it can be traced back to where it was injected so that identifying what changes need to be made is easier.

Changes to the requirements, while not ideal, sometimes are necessary in order to adapt to identified and unavoidable risks in the project, changes in the customer’s expectations and needs to the proposed product or slips in the project schedule which require a scaling down of the requirements. These changes will be supported by sufficient rationale for the necessity of requirement alteration for the success of the project. The proposed changes will be reviewed by the project authority for necessity and project impact. If accepted, the SRS will be altered as per the documentation and configuration processes defined in section 6. Design and implementation choices linked to the altered requirement will similarly be updated.

#### 5.3.2 Schedule Tracking and Adjustment

The project schedule plan will be maintained in the form of a Gantt Chart made using Microsoft Excel. The project schedule is also maintained within a shared Google Calendar amongst the project team. Completion of various milestones and work activities are recorded and marked as such in the Google Calendar. Progress made in each work activity is measured based upon the completion of the various goals and supporting processes of each work activity as defined previously. An activity is considered complete if all of its output were generated at the specified level of quality.

Based upon the progress made towards each work activities by the various schedule milestones, the schedule will be periodically revisited to assess its accuracy. Schedule slips may require that the schedule be refactored in order to accommodate needed milestone and timeline extensions. Such extensions, if they interfere with the deadline imposed by the project authority, the schedule revision will be discussed with the project authority. Due to the short timeframe allotted for the project, schedule slips of greater than 3 days will be marked as needing corrective action by the project team and manager. Cumulative schedule slips (i.e. the actual progress is three days later than the planned scheduling) of less than 3 days will be considered minor and not warranting a change in the schedule.

#### 5.3.3 Budget Tracking and Adjustment

Not included in version 1 of this document.

#### 5.3.4 Quality Control

The quality of work products will be ensured by a combination of reviews and checks. The quality control plan is included in section 7.2.

#### 5.3.5 Reporting Mechanisms

Not included as of version 1 of this document.

#### 5.3.6 Metrics Collection Plan

The primary metrics to be collected and summarized in each project document are:

* Number of defects in project artifacts
* Time spent on each work activity / project artifact

These metrics are summarized in the appendix of this document. The number of defects are a reflection of the quality of the project. Time spent on working on each activity and project artifact are useful towards the schedule management and allocation processes of the project. Metrics are estimated prior to the start of each work activity then compared to the actual number as collected by the project authority and project team.

### 5.4 Risk Management Plan

Risk management is performed throughout the project lifecycle. Risks are classified as either technological risks (i.e. risks associated with the technical development of the project), environmental risks (i.e. risks associated with the factors external to the project), economic risks (i.e. risks associated with the business aspects of the project), and operational risks (i.e. risks associated with the management of the project). Risks are continually tracked in a risk matrix which identifies the risk classification, the risk and description of it, probability of occurrence, the party responsible for its resolution, severity, discovery date, mitigation plan, and current status of the risk. After each milestone is submitted, the project team reviews the risk classifications from previous stages of the project and updates the risk matrix accordingly. Furthermore, risk management activities are assigned to the various responsible parties within the project team based upon the mitigation strategies defined upon risk discovery. Furthermore, new risks are identified given the progress and development of the project.

### 5.5 Post Implementation Plan

Not included in Version 1 of this document.

## 6 TECHNICAL PROCESSES

### 6.1 Process Model

The project team will follow the Waterfall Process through the first two versions of the project. More specifically, the first and second version of Feed Me will consist of at least, Authentication and Register components. Afterward, the members will follow the Iterative approach, more precisely an Agile Process.

Each iteration will result in the release of a major component. The first iteration begins after Authentication and Register components are developed (including documentation), tested and tested via Waterfall process. Afterward, the team will adhere to a code-oriented paradigm with numerous components being developed in parallel.

The master copy of codebase for the project must reside on a distributed code repository which is accessible to each team member. The training and testing data for machine learning components must reside on cloud-based storage which is accessible to every team member.

Each member is allowed to commit only style asserted codebase to the repository. Artifacts which are yet to be tested must be committed to Upstream branch. Each iteration will result in merging Upstream into the Master branch. It implies that the Master branch is the up-to-date product available for presentation. Temporary components will be either stashed or committed to the component-specific branch.

### 6.2 Methods, Tools and Techniques

The team will utilize the following methods:

1. Documenting On As-Needed basis;
2. Pair Programming;
3. Refactoring.

Documenting on As-Needed basis is enforced by each deadline imposed by the Software Development class.

Pair Programming will assure the artifact will be delivered in a timely manner with satisfactory quality. The team for Pair Programming will consist of two members; one member will devise test cases for the artifact while the second team member continues in development. The third member of the team will hold the “floater” title. The floater will either focus on refactoring developed artifacts or will start developing demanded artifact.

Refactoring will be the main task of “floater” team member. The purpose behind the method is to simplify artifact’s codebase without losing its functionality as well as quality. The refactoring method will be solely imposed on already tested and from major part developed artifacts.

This project shall use the following programming tools:

* **Back-End Programming** will be developed in Python v3.7.2 using open-source Python-Flask v1.0.2 package. In conjunction with the Python-Flask codebase, the team will be using the open-source MySQL v5.1 database with Python-Flask compatible extension known as SQLAlchemy v1.2.19. Finally, the machine learning part of the application will be trained and updated via Sk-Learn Python package.
* **Front-End Programming** Tools will consist of HTML5, CSS3, React JavaScript Framework in addition to JavaScript. These programming tools will be solely for the purpose of developing artifacts residing on client’s machine.
* **Documentation** will be generated through Python Doc where available.
* **Programming Style** will be asserted via ESLint tool for both Front-End and Back-End programming.

### 6.3 Infrastructure Plan

The minimum infrastructure for the project:

* Cloud based memory storage;
* Decentralized codebase repository;
* Configuration files for shared development server.

### 6.4 Product Acceptance and Migration Plan

Not included in Version 1 of this document.

## 7 SUPPORTING PROCESSES PLAN

### 7.1 Configuration Management Plan

Configuration management is the process by which development items (both components of the final product and documentation) are developed, baselined, controlled and then updated during the project lifecycle. Potential configuration items to be managed are:

* Software components (source code, libraries, and compilers)
* Design documentation (API references and development note)
* Project documentation (documentation on project processes such as this SPMP, SRS, Project Proposal, etc.)

The project team will use GitHub as its configuration management system. A single organization will be created (called “Feed Me” on GitHub) and the developers on the project team will be added to the organization as collaborators. All changes made to the configuration items will be reviewed by the development team (using the quality assurance plan). Repositories will be divided based upon the product it is tracking (similar components of the project such as the server components, client components, documents, etc. will be grouped into the same repository).

Each developer will maintain their own branch in each repository and commit their changes to said branch. Upon completion of the review processes, baselined products in individual branches will be merged into the master branch. The resolution of merge and branch conflicts will be handled by the configurational manager (see roles and responsibilities).

### 7.2 Qualification (Verification and Validation) Plan

The test plan’s purpose is to ensure that the final system conforms to the requirements previously defined so that it meets the needs and expectations of the customers and stakeholders. The most effective way of doing so would be to test the product within a closed group of testers who belong to the target audience / user-base. Therefore, the requirement for the testing of the final system should include the following components:

* The testers must belong to the identified user groups from and exhibit the characteristics described in Section 3.2.2. This is to ensure that the testing closely resembles the wild usage of the product after release.
* The testers must be allowed full use of the platform in its production state and the developers / testers should not interfere with their usage of the platform. This requirement includes the prohibition of developers from being able to explain the platform to the testers as they would not have said interaction available after the system is deployed to actual users.

The requirements will be delivered to SQA who will create a test plan including test scenarios and expected output, then they will execute the tests and report defects.

### 7.3 Documentation (Library) Plan

All the documentation for the project will be maintained in the configuration management system as a separate repository of the GitHub organization hosting the configuration items as baselined configuration items. Collaboration will occur using Google Docs to facilitate simultaneous editing and version control. Upon final release of a document’s major version, it will be exported from Google Docs and into the GitHub repository.

### 7.4 Quality Assurance Plan

The quality of this and future artifacts will be ensured by various quality testing procedures. These procedures will be checking for the following metrics:

* The artifact will be reviewed to ensure that it reflects the original purpose of the project as defined in the previous documents by the project team before submission for review (**correctness**)
* The artifact will be reviewed to ensure **unambiguity** (a potential software developer interprets the document is it should be)
* The artifact will be compared to the original document standard to ensure that all required components are included in the document (**completeness**).
* The artifact will be examined for **consistency** between itself and previous artifacts generated by the project team. For example, the wording and implications of requirements must be maintained throughout all of the documents and artifacts when specified.
* The artifact will be examined for its **verifiability** and **traceability**. In other words, can the statements being made in the document be both seen and checked for in the final product?
* The artifact will be examined for its **malleability**. In the event of a risk manifesting or an assumption failing, can the content of the document be changed to respond to external changes?

Four types of testing will be performed: a desk check, a peer review, a walkthrough and an inspection. The first method of ensuring the quality of this artifact, a desk check, essentially involves the writers of the document proofreading their work such that they identify errors they made when drafting the artifact such as typos, obscurities and confused wording. A peer review involves team members reviewing each other’s work to identify possible defects and errors made during the drafting process. A peer review allows for more individuals to review the work and helps to improve the clarity of the artifact as individuals evaluating their own work tend not to see their own ambiguities as they are familiar with their own thought process.

Walkthroughs involve a group of reviewers having the artifact author “walkthrough” their work in person in front of the reviewers (in this case the rest of the project team and the SQA) so that the reviewers can check that the artifact adequately conveys the author’s intent. Finally, inspections involve submitting the artifact to an independent review panel for review. In the case of this document, it is submitted to the SQA and project authority for review.

### 7.5 Reviews and Audits

For the schedule of the reviews conducted for the project (desk checks, peer reviews, walkthroughs and inspections) see the Gantt Chart in the appendix of this document.

Essentially, each deliverable will be reviewed following the completion of initial development. The developer will be responsible for performing a desk check prior to commission of their work to the configuration management workflow. Following commission (and generally after the other developers are have committed their work), the other developers will perform a peer review on the submitted artifact (or portion of an artifact). The developers would make the necessary corrections to their documents based upon the peer reviews. Comments and questions indicated during the peer review will be addressed during the walkthrough when, in person, each developer explains their work in person and answers any necessary questions regarding their work. Changes made following the walkthrough will be submitted to the project authority for an inspection.

The final inspected product will be baselined and merged into the master branch of the configuration management repository.

Other review processes will occur on the management aspects of the project. For example, periodically, the project team will meet to assess actual progress made on the project as compared with the previously allocated schedule. Small slips may be ignored; however, large set-backs in the project may result in alterations being made to the project. Significant project delays may require the project team meeting with the project authority to reassess the feasibility of the project and its timeline as well as alternatives to be taken to recover to the original schedule.

### 7.6 Problem Resolution Plan

Project problems are tracked in an issue log. Problems differ from risks as risk are usually identified before they occur in the project (and are therefore avoidable) whereas problems are usually encountered as they occur and therefore must be recovered from rather than prevented. Problems can include uncaught defects in an artifact, unforeseen schedule slips, personnel issues, or unforeseen development blockers (such as a service shutting down to which the project has a dependency).

Problems are tracked in a problem log, which contains when the problem was encountered, the location of the problem, a description of the problem, its severity, its priority (how soon should the problem be addressed), how the problem will be resolved, and its current status. High priority problems are addressed as soon as the problem is encountered (developers shall move resolution of the problem to the top of their task list), while lower priority problems will be addressed as developers become available to do so.

Generally, problem discovery will occur during the quality assessment process during reviews of documents. Problems caught and logged in this process may be resolved immediately during revision of artifacts after each review stage. Other problems may occur during configuration management (such as branch conflicts) or change control (merge conflicts). Such problems will be managed by the configurational manager who will work with developers to both trace the problem, them take action to resolve it.

### 7.7 Environmental Management Plan

Not included in Version 1 of this document.

### 7.8 Process Improvement Plan

Not included in Version 1 of this document

## 8 ADDITIONAL PLANS

Not included in Version 1 of this document.

## 9 INDEX

Nothing as of Version 1 of this document.

## 10 RATIONALE

Nothing as of Version 1 of this document.

## 11 NOTES

Nothing as of Version 1 of this document.

## 12 APPENDICES

### 12.1 Schedule Tracking

TABLE 12.1-1

Individual Artifact Schedule Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or team) | Estimated | Actual | Difference |
| SRS – Business Definition | Theodore Kim | 5 hours | 6 hours | +1 hour |
| JinZhao Su | 3 hours | 2 hours | -1 hour |
| Petr Holoubeck | 3 hours | 3.5 hours | +0.5 hours |
| Entire Team | 11 hours | 11.5 hours | +0.5 hours |
| SRS - Requirements | Theodore Kim | 5 hours | 4 hours | -1 hour |
|  | JinZhao Su | 2 hours | 2 hours | 0 hours |
|  | Petr Holoubeck | 2 hours | 4 hours | +2 hours |
|  | Entire Team | 9 hours | 10 hours | +1 hour |
| SPMP | Theodore Kim | 8 hours | 12 hours | +4 hours |
|  | JinZhao Su | 5 hours | 6 hours | +1 hour |
|  | Petr Holoubeck | 6 hours | 8 hours | +2 hours |
|  | Entire Team | 19 hours | 26 hours | +7 hour |

TABLE 12.1-2

Cumulative Schedule Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or team) | Estimated | Actual | Difference |
| Theodore Kim | 18 hours | 22 hours | +4 hours |
| JinZhao Su | 10 hours | 13.5 hours | +3.5 hours |
| Petr Holoubeck | 11 hours | 12 hours | +1 hour |
| Entire Team | 39 hours | 47.5 hours | +8.5 hours |

### 12.2 Defect Tracking

TABLE 12.2-1

Individual Artifact Defect Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or team) | Estimated | Actual | Difference |
| SRS – Business Definition | Theodore Kim | 3 defects | 5 defects | +2 defects |
| JinZhao Su | 4 defects | 3 defects | -1 defects |
| Petr Holoubeck | 3 defects | 2 defects | -1 defects |
|  | Entire Team | 10 defects | 10 defects | 0 defects |
| SRS – Requirements | Theodore Kim | 5 defects | 3 defects | -2 defects |
|  | JinZhao Su | 3 defects | 0 defects | -3 defects |
|  | Petr Holoubeck | 6 defects | 2 defects | -4 defects |
|  | Entire Team | 14 defects | 5 defects | -9 defects |
| SPMP | Theodore Kim | 3 defects |  |  |
|  | JinZhao Su | 2 defects |  |  |
|  | Petr Holoubeck | 3 defects |  |  |
|  | Entire Team | 8 defects |  |  |

TABLE 12.2-2

Cumulative Schedule Defect Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or team) | Estimated | Actual | Difference |
| Theodore Kim | 8 defects | 8 defects | 0 defects |
| JinZhao Su | 7 defects | 3 defects | -4 defect |
| Petr Holoubeck | 9 defects | 4 defects | -5 defect |
| Entire Team | 24 defects | 15 defects | -9 defects |

### 12.3 Project Schedule Gantt Chart

