project management plan

for the

Statusphere Project

version 1.0

February 9, 2017

***Place project logo here***

Project Owner: Kristen Wiley

Project Manager: Brandon Jones

Statusphere Front-end Development Team:

|  |  |
| --- | --- |
| Amith Lakshmisha | Arpitha Rajanna |
| Sukmin Kim | Leena Jawale |
| Sathya Narayanan Amarnath | Pavitra Shivanand Hiremath |
| Pooja Shivashankar | Nithyashree Kempegowda |
| Swathi Kanthareddy Sumithra |  |

Statusphere Back-end Development Team:

|  |  |
| --- | --- |
| Nishant Sahni | Somnath Saha |
| Atul John | Gaurav Kataria |
| Soham Ganguly | Xuejiao Yang |

[[Project Name, Code]]

Preface

This Project Management Plan (PMP) is intended to provide guidance on the management of the Statusphere.

The template conforms to the Institute of Electrical and Electronics Engineers (IEEE) Standard for Software Project Management Plans, IEEE Std 1058-1998, for format and content. The template and its standard were selected as they are flexible enough to be applied to any type of project. The management, technical, and supporting processes comply with the guidance provided by Standard for Information Technology - Software Life Cycle Processes, IEEE/Electronic Industries Association (EIA) 12207 Series; Systems Engineering – System Life Cycle Processes, International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 15288; or the Processes for Engineering a System, Electronic Industries Alliance (EIA) Standard 632.

DOCUMENT CONVENTIONS

The outline of this Project Management Plan (PMP) has been tailored from the Institute of Electrical and Electronics Engineers (IEEE) Standard for Software Project Management Plans, IEEE Std 1058-1998.

Standard conventions are used within this document to direct the reader to specific sections of the text. These sections provide instructions and explanations and require users to insert their own project-specific information. The conventions used in this document are shown below.

[[text]] Global changes. Items that appear in regular text and are surrounded by double brackets represent changes that can be made globally throughout the document.

*Italics* Instructions and explanations. Each section of the template has been annotated with a guidance box, derived from the IEEE 1058-1998 standard, to assist the reader in drafting the content. For example:

***IEEE Std 1058-1998 Guidance***

The guidance box provides instructions and explanations from the IEEE 1058-1998 Standard, in italics, as required to assist the user in drafting their own information.

Guidance boxes should be deleted from the final PMP.RECORD OF CHANGES

\*A - ADDED M - MODIFIED D – DELETED

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VERSION NUMBER | DATE | NUMBER OF FIGURE, TABLE OR PARAGRAPH | **A\* M D** | TITLE OR BRIEF DESCRIPTION | CHANGE REQUEST NUMBER |
| 1.0 | 2/9/17 |  | A\* | Initial Version |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

Section Page

[SECTION 1. OVERVIEW 1](#_Toc257388665)

[1.1 Project Summary 1](#_Toc257388666)

[1.1.1 Purpose, Scope, and Objectives 1](#_Toc257388667)

[1.1.2 Assumptions and Constraints 1](#_Toc257388668)

[1.1.3 Project Deliverables 1](#_Toc257388669)

[1.1.4 Master Schedule and Budget Summary 1](#_Toc257388670)

[1.2 Evolution of the Plan 2](#_Toc257388671)

[1.3 Document Structure 2](#_Toc257388672)

[SECTION 2. REFERENCES 4](#_Toc257388673)

[2.1 Standards and Documents 4](#_Toc257388674)

[SECTION 3. DEFINITIONS 5](#_Toc257388675)

[SECTION 4. PROJECT ORGANIZATION 6](#_Toc257388676)

[4.1 External Interfaces 6](#_Toc257388677)

[4.2 Internal Structure 6](#_Toc257388678)

[4.2.1 The Project Manager 6](#_Toc257388679)

[4.3 Project Roles and Responsibilities 6](#_Toc257388680)

[SECTION 5. MANAGEment PROCESS 7](#_Toc257388681)

[5.1 Start-up 7](#_Toc257388682)

[5.1.1 Estimation 7](#_Toc257388683)

[5.1.2 Staffing 7](#_Toc257388684)

[5.1.3 Resource Acquisition 7](#_Toc257388685)

[5.1.4 Staff Training 8](#_Toc257388686)

[5.2 Work Planning 8](#_Toc257388687)

[5.2.1 Work Activities 8](#_Toc257388688)

[5.2.2 Schedule Allocation 8](#_Toc257388689)

[5.2.3 Resource Allocation 9](#_Toc257388690)

[5.2.4 Budget Allocation 9](#_Toc257388691)

[5.3 Project Controls 9](#_Toc257388692)

[5.3.1 Requirements Control 9](#_Toc257388693)

[5.3.2 Schedule Control 10](#_Toc257388694)

[5.3.3 Budget Control 10](#_Toc257388695)

[5.3.4 Quality Control 11](#_Toc257388696)

[5.3.5 Project Reporting and Communication 11](#_Toc257388697)

[5.3.6 Metrics Collection 11](#_Toc257388698)

[5.4 Risk Management 12](#_Toc257388699)

[5.5 Project Closeout 12](#_Toc257388700)

[SECTION 6. TECHNICAL PROCESS 13](#_Toc257388701)

[6.1 Process Model 13](#_Toc257388702)

[6.2 Methods, Tools and Techniques 13](#_Toc257388703)

[6.3 Project Infrastructure 13](#_Toc257388704)

[6.4 Product Acceptance 13](#_Toc257388705)

[SECTION 7. SUPPORTING PROCESSES 15](#_Toc257388706)

[7.1 Configuration Management 15](#_Toc257388707)

[7.2 Independent Verification and Validation 15](#_Toc257388708)

[7.3 Documentation 15](#_Toc257388709)

[7.4 Quality Assurance 16](#_Toc257388710)

[7.5 Reviews and Audits 16](#_Toc257388711)

[7.6 Problem Resolution 16](#_Toc257388712)

[7.7 Contractor Management 16](#_Toc257388713)

[7.7.1 Contracting Process 17](#_Toc257388714)

[7.7.2 Contractor Performance Monitoring 17](#_Toc257388715)

[7.8 Process Improvement 17](#_Toc257388716)

[7.8.1 Systems/Software Process Improvement Lead 17](#_Toc257388717)

[7.8.2 Systems Engineering Process Group 17](#_Toc257388718)

[SECTION 8. Additional Plans 18](#_Toc257388719)

[Appendix A. StatusphereMaster Schedule (Microsoft Project) 2](#_Toc257388720)

[Appendix B. StatusphereFacilities Plan 3](#_Toc257388721)

[Appendix C. StatusphereProject Training Plan 4](#_Toc257388722)

[Appendix D. Statusphere Measurement Plan 5](#_Toc257388723)

[Appendix E. Statusphere Product Engineering and Qualification Process 6](#_Toc257388724)

[Appendix F. StatusphereQuality Assurance Plan 7](#_Toc257388725)

[Appendix G. StatusphereConfiguration Management Plan 8](#_Toc257388726)

List of Figures

Figure Page

List of Tables

Table Page

[Table 7-1. Statusphere Documentation 16](#_Toc115065445)

# SECTION 1. OVERVIEW

## Project Summary

The team will be developing a system, Statusphere, for providing select social media *Influencers* with advertising opportunities; this entails: building a front-end, both a web portal and a mobile app, for influencers to select advertising *packages*; building a front-end for administrative tasks (such as accepting new influencers and adding new advertising packages); building a back-end for storing both influencer and advertiser package information; and, developing an automated acceptance system for selecting influencers.

### 1.1.1 Purpose, Scope, and Objectives

The system, Statusphere, involves an Administrator, who maintains and manages the profiles of social media Influencers, who advertise the products in respective social media applications as per the requirements of the administrator. The objective of the influencer is to accept a product package from the administrator and publicize them in social media so that people will be influenced by the post. The system uses web and mobile applications for managing the portal of administrator and influencers including backend technologies for storing their details and automating and performing data analytics. Satisfaction will be determined by both the product owner (Kristen Wiley) and select test users. This project will replace existing web services provided by third-party entities.

### 1.1.2 Assumptions and Constraints

The assumptions and constraints in this project are:

* The project may be developed for both Android and iOS applications.
* Various platforms required include- AngularJS, jquery, Bootstrap, Django or flask for web development, Android studio for android app, swift or objective c for iOS and few backend technologies for data analytics.
* The project has two teams – one for front end design and another one for backend.
* Front end team consist of 10 members and back end consist of 6 members.
* Each member in the teams have sufficient technical competency; but, might require training for some technologies.
* Each member is ready to work 20 hours a week, unless overtime is required to keep the project on schedule.
* Replacement team members are not available; although, team assignments can be changed to fit the workload.
* The duration of the project is limited to the UCF Spring 2017 semester.

### 1.1.3 Project Deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Delivery Date** |
| **Statement of Work** | Jan. 26 |
| **Project Management Plan** | Feb. 9 |
| **Software Requirements Specification** | Feb. 16 |
| **Software Design Description** | Mar 2. |
| **Peer Review Evaluations** | Mar. 9 |
| **Software Test Plan** | Mar. 21 |
| **Progress Review** | Mar. 30 |
| **Software Acceptance Tests** | Apr. 20 |

### 1.1.4 Master Schedule and Budget Summary

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Additional Details** | **Delivery Date** |
| Statement of Work |  | Jan. 26 |
| Initial Requirements Elicitation |  | Jan. 27 |
| Initial Requirements Validation | Priority of requirements was determined. Begin 1st sprint. | Feb. 2 |
| Project Management Plan |  | Feb. 9 |
| Software Requirements Specification / High-level Design Review /  **Milestone: mock interface prototypes** | Begin low-level/component design and development. Being 2nd sprint. **Present mock *Influencer* interface prototypes.** | Feb. 16 |
| Software Design Description / Low-level Design Review | Begin component development. Begin 3rd sprint. | Mar 2. |
| Peer Review Evaluations |  | Mar. 9 |
| Progress Review /  **Milestone: partially-functional prototypes** | Begin 4th sprint. **Present (partially-functional) interface prototypes.** | Mar. 16 |
| Software Test Plan |  | Mar. 21 |
| Progress Review | Begin 5th sprint. **Present functional system prototype.** | Mar. 30 |
| Progress Review | Begin final sprint; freeze requirements. | Apr. 13 |
| Software Acceptance Test |  | Apr. 20 |

## 1.2 Evolution of the Plan

TBD.

## 1.3 Document Structure

This plan is organized as follows:

1. Section 1, Project Overview. This section provides an overview of the scope and objectives of the project, the project’s assumptions and constraints, reference to the project deliverables, schedule and budget, and a description of the evolution of the plan.
2. Section 2, References. This section provides a list of all documents, policies, templates, processes, and other sources of information referenced in the plan.
3. Section 3, Definitions. This section contains the abbreviations and acronyms required to properly understand this planning document.
4. Section 4, Project Organization. This section identifies interfaces to organizational entities external to the project, the project’s internal organizational structure, and defines roles and responsibilities for the project.
5. Section 5, Management Process. This section describes the planning, measurement, tracking, reporting, risk control mechanisms needed to provide management control over the technical processes and product quality, and appropriate project initiation and closeout procedures.
6. Section 6, Technical Process. This section describes the technical solution in terms of a process model and implementation methods, tools, and techniques to be used to develop the various work products, plans for establishing and maintaining the project infrastructure, and the product acceptance.
7. Section 7, Supporting Processes. This section describes processes that are employed to facilitate and control the technical processes and the state of the product. These include, but are not limited to, configuration management, verification and validation, documentation, quality assurance, reviews and audits, problem resolution, and contractor management, and methods to ensure continuous process improvement.
8. Section 8, Additional Plans. This section addresses the logistic support strategy to be applied to increase the system’s operational effectiveness.
9. Appendix A. StatusphereMaster Schedule (Microsoft Project)
10. Appendix B. StatusphereFacilities Plan
11. Appendix C. StatusphereProject Training Plan
12. Appendix D. StatusphereMeasurement Plan
13. Appendix E. StatusphereProduct Engineering and Qualification Process
14. Appendix F. StatusphereQuality Assurance Plan
15. Appendix G. StatusphereConfiguration Management Plan

# SECTION 2. REFERENCES

## 2.1 Standards and Documents

The standards and documents listed below are referenced in this document:

# SECTION 3. DEFINITIONS

TBD: To be determined

# SECTION 4. PROJECT ORGANIZATION

## 4.1 External Interfaces

The parent organization is Joinstatus, with CEO Kristen Wiley acting as the interface between the team and the organization. Dr. Edwin Nassiff will also serve as a project coordinator.

## 4.2 Internal Structure

### 4.2.1 The Project Manager

The project manager is Brandon Jones. The primary customer (also referred to as the Project Owner) is Kristen Wiley, of JoinStatus; additionally, Dr. Edwin Nassiff will assume the role of a customer.

**4.2.1.1 Scope of Authority**.

The project manager will have authority over the assignment of all work tasks to all members of the team, as well as all design decisions; and, the authority to delegate portions of this role to other individuals. The scope of the project manager’s authority may be changed at any time at the discretion of the customers.

**4.2.1.2 Scope of Responsibility**.

The project manager will be responsible for the delivery of a satisfactory project within the fixed duration of the project. Specifically, the project manager will be responsible for: the assignment of high-level work tasks; the facilitation of communication across teams; arranging project meetings – additional meetings might be arranged for each team by the teams’ respective members; the compilation and management of all documentation; and the resolution of technical and/or personal disagreements between team members.

## 4.3 Project Roles and Responsibilities

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Team** | **Name** | **Planning** | **High-Level Design** | **Detailed Design** | **Implementation** | **Unit Testing** | **Integration Testing** | **Documentation** |
| **Project Manager** | Brandon Jones | P | P | TBD | TBD | TBD | TBD | P |
| **Front-End** | Amith Lakshmisha | S | S | TBD | TBD | TBD | TBD | N |
| Sukmin Kim | S | S | TBD | TBD | TBD | TBD | N |
| Sathya Narayanan Amarnath | S | S | TBD | TBD | TBD | TBD | N |
| Pooja Shivashankar | S | S | TBD | TBD | TBD | TBD | N |
| Swathi Kanthareddy Sumithra | S | S | TBD | TBD | TBD | TBD | N |
| Arpitha Rajanna | S | S | TBD | TBD | TBD | TBD | N |
| Leena Jawale | S | S | TBD | TBD | TBD | TBD | N |
| Pavitra Shivanand Hiremath | S | S | TBD | TBD | TBD | TBD | N |
| Nithyashree Kempegowda | S | S | TBD | TBD | TBD | TBD | N |
| **Back-end** | Nishant Sahni | S | S | TBD | TBD | TBD | TBD | N |
| Atul John | S | S | TBD | TBD | TBD | TBD | N |
| Soham Ganguly | S | S | TBD | TBD | TBD | TBD | N |
| Somnath Saha | S | S | TBD | TBD | TBD | TBD | N |
| Gaurav Kataria | S | S | TBD | TBD | TBD | TBD | N |
| Xuejiao Yang | S | S | TBD | TBD | TBD | TBD | N |

P=Primary responsibility

S=Secondary responsibility

N=Not responsible

# SECTION 5. MANAGEment PROCESS

## 5.1 Start-up

The following sections comprise the project start-up plan: estimation plan, staffing plan, resource acquisition plan and training plan.

### 5.1.1 Estimation

|  |  |  |
| --- | --- | --- |
| Work Activity | Person Hours | % Effort |
| Project Management Plan | 40 | 2.78% |
| Software Requirements Specifications | 120 | 8.33% |
| Software Design Specification | 150 | 10.42% |
| Software Test Plan | 150 | 10.42% |
| Research and Training | 320 | 22.22% |
| Back-End Framework | 120 | 8.33% |
| Web Development | 120 | 8.33% |
| iOS Development | 200 | 13.89% |
| Android Development | 100 | 6.94% |
| Unit testing | 80 | 5.56% |
| System Testing | 40 | 2.78% |
| **Total** | 1440 | 100.00% |

### 5.1.2 Staffing

This project is limited to the 16 assigned members; all members are expected to participate for the duration of the UCF Spring 2017 semester. The skill level of each team member, language and development skill is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Java** | **Python** | **C/C++** | **iOS Dev.** | **Android Dev.** | **Web Dev.** | **HTML** | **Javascript** | **AngularJS** | **Node.js** | **Security** | **Databases** |
| Sukmin | Front | 4 | 3 |  | 3 | 5 |  | 4 |  |  |  | 3 |  |
| Sathya | Front | 3 |  | 3 |  |  | 3 |  |  |  |  |  | 4 |
| Leena | Front | 3 |  | 3 |  |  | 4 |  | 3 |  |  |  | 4 |
| Nithya | Front | 2 |  | 3 |  | 1 |  | 3 |  |  |  |  |  |
| Swathi | Front | 2 |  | 3 |  | 2 |  | 3 |  |  |  |  | 2 |
| Arpitha | Front | 1 |  | 4 | 1 | 1 |  | 2 |  |  |  |  | 2 |
| Pooja | Front | 2 |  | 3 |  | 1 |  | 2 |  |  |  |  |  |
| Pavitra | Front |  |  | 3 |  |  |  | 2 |  |  |  | 3 |  |
| Somnath | Back | 3 | 4 | 4 |  | 3 | 3 | 4 |  |  |  |  |  |
| Nishant | Back | 3 | 3 | 3.5 |  | 2 |  |  |  |  |  |  | 1 |
| Xuejiao | Back |  | 4 | 2 |  |  |  |  |  |  |  |  | 2 |
| Guarav | Back | 4 | 4 |  |  | 4 | 3 |  |  |  |  |  | 3.5 |
| Atul | Back | 3 |  | 4 |  |  |  |  |  |  |  |  |  |
| Amith | Front | 3 | 3 | 4 | 2 | 3 |  |  |  |  |  |  |  |
| Soham | Back | 4 |  | 4 |  |  | 2 |  | 3 |  |  |  | 3 |

### 5.1.3 Resource Acquisition

TBD.

### 5.1.4 Staff Training

Individuals will seek training on their own, as necessary.

## 5.2 Work Planning

The following paragraphs provide a working management plan for the acquisition of the Statusphere.

### 5.2.1 Work Activities

TBD.

### 5.2.2 Schedule Allocation

TBD.

### 5.2.3 Resource Allocation

### TBD.

### 5.2.4 Budget Allocation

TBD.

## 5.3 Project Controls

### 5.3.1 Requirements Control

TBD.

### 5.3.2 Schedule Control

The following paragraphs define the management approach for schedule control of the Statusphere.

**5.3.2.1 Schedule Tracking**.

Schedules will be proposed before any group meetings and documented on Slack; any changes to the schedule will be indicated in the post for the proposed meeting. Schedules for component design, implementation, and testing, will be tracked by each team separately.

**5.3.2.2** **Schedule Performance Reports**.

TBD.

**5.3.2.3 Schedule Reviews**.

TBD.

**5.3.2.4 Progress Variance Monitoring**.

TBD.

**5.3.2.5 Progress Variance Resolution**.

TBD.

**5.3.2.6 Follow‑Up on Corrective Action**.

### TBD.

### 5.3.3 Budget Control

The following paragraphs define the management approach for schedule control of the Statusphere.

**5.3.3.1 Cost Management**.

The team members will document their hours worked each week in the team member timesheet; the project manager will compile the weekly timesheets.

**5.3.3.2 Methods to Ensure Cost Adherence**.

The project manager will conduct one-on-one meetings with team members that are either working a significantly below-average number of hours (the appropriate number of work hours will be determined on a weekly basis) or are delivering unsatisfactory work items.

**5.3.3.3 Cost Control**.

TBD

**5.3.3.4 Contractor Cost Control**.

The project does not require any work by external contractors.

**5.3.3.5 Cost Variance Measurement**.

TBD

**5.3.3.6 Cost Variance Corrective Action**.

TBD

### 5.3.4 Quality Control

Quality control is an integral part of the project to assure that the project satisfies the needs for which it was undertaken. The work plan of the project describes milestones and the acceptance criteria for each phase of the project. The following quality control activities will be conducted to measure and control the quality of the work processes and the resulting work products:

* Internal validation: Internal validation will occur through peer reviews. All the artifacts including source code, documents, diagrams etc. will be maintained under GitHub. This provides facilities for all the team members to review the items and provide comments on each other’s work. Programmers would be asked to review other programmer’s code.
* External validation: Customer satisfaction is an important indicator of the quality of the product. Owner of the project is involved throughout the life of the project in establishing requirements, accepting deliverables and accepting products of the project. Their responses provide additional data for evaluating management of the project and how successfully the project achieves its purpose. Customer reviews to validate acceptance and customer satisfaction for each milestone and for the acceptance criteria will be carried out frequently if required.

### 5.3.5 Project Reporting and Communication

The following paragraphs define the management plan for ensuring the broadest communication of needed information for project coordination.

**5.3.5.1 Electronic Media**.

All team members will use UCF Knight’s email and slack for electronic communication.

**5.3.5.2 Meetings**.

Meetings will be held once a week to ensure that the project deadlines will be met. Bi-weekly meetings will be held with the project owner to ensure adherence to requirements and to ensure the quality of deliverables.

**5.3.5.3 Information Repository**.

The source code and documentation will be stored and tracked on GitHub and Slack, respectively.

**5.3.5.4 Reviews**.

Each team will review completed work items during separate weekly meetings; the project manager will review completed work items at weekly team meetings, or more frequently, as necessary. The project manager will conduct one-on-one reviews with under-performing team members twice a week, until performance is deemed to be satisfactory again.

**5.3.5.5 Status Reporting**.

### Status will be reported during weekly team meetings and bi-weekly meetings with the project owner.

### 5.3.6 Metrics Collection

TBD.

## 5.4 Risk Management

TBD.

## 5.5 Project Closeout

TBD.

# SECTION 6. TECHNICAL PROCESS

## 6.1 Process Model

The team shall follow an agile process with biweekly or weekly sprints and daily scrums. The frequency of the sprints and scrums may change based on the need and situation as the program progresses. Since the team is large, a daily scrum meeting might be beneficial. The basic stages of Agile process are Planning, Requirement analysis, Designing, Building and Testing. Each sprint will end with an incremental deliverable to the customer.

It is important to stick to the sprint deadlines as the customer must see functionality being added. Constant feedback is required to make sure the team implements features as expected by the customer. Since Agile process does not involve documentation, this could lead to a problem in the future in terms of tracking features and individual contributions. It might be wise to have a barebones documentation to avoid this issue.

## 6.2 Methods, Tools and Techniques

Since we are building web based applications, we can utilize various web frameworks available online instead of having to write from scratch. The team is considering the following frameworks.

### 6.2.1 Web Technology

1. Bootstrap (http://getbootstrap.com/getting-started/)
2. JQuery (<https://jquery.com/)>
3. AngularJS (<https://angularjs.org/)>
4. Django (https://www.djangoproject.com/start/) [Python based] 5. Flask (http://flask.pocoo.org/) [Python based]

### 6.2.2 Android

Java Programming using Android Studio/Eclipse.

### 6.2.3 iOS

Swift.

### 6.2.4 Hybrid Mobile Frameworks

The team is also considering frameworks in which Android and iOS apps can be created without writing separate code for each OS; such as OnsenUI or Ionic, combined with Cordova (a.k.a. Phonegap).

### 6.2.5 Development Tools

GitHub will be used for version control. All team members must create a GitHub account and must use it for downloading and committing code. This will ensure we have a unified code base and more control over the code. The programming languages used will be HTML, JavaScript, Python, Java (for Android), and Swift (for iOS).

## 6.3 Project Infrastructure

The major infrastructure will be web space. Instead of using a hosting site, we will be using the Cloud. We expect to run all our applications on the cloud, particularly AWS as the customer has $5000 credit from AWS. Also, moving to the cloud allows for greater scalability.

Depending on the framework chosen, an instance of the AWS machines will run as the server. Each person in a development team will have a sandbox on his/her local machine where they will have the latest copy of the code from GitHub and whatever changes they have made locally.

## 6.4 Product Acceptance

The product will be accepted if it is able to assist in the handling of applicant users; maintain and adequately display available packages to users; assist in the creation of packages for users to view and select; and, assist in the tracking of package completion. Additionally, the aesthetic quality and ease-of-use of the system must be determined to be adequate by the project owner.

# SECTION 7. SUPPORTING PROCESSES

## 7.1 Configuration Management

The following documents will be maintained under configuration management:

* UML diagrams
* Design documents
* Software risk analysis documentation
* Source code documentation
* Database documentation
* Software testing documentation
* Software verification documentation
* Testing data; real user data will not be maintained under configuration management.

The following activities will be performed under configuration management:

* Source code will be properly documented.
* Proper version control will be implemented.
* Output generated from various tasks will be used by the team.
* Testing will be performed every time the new version of the source code will be released.
* Relations between various classes will remain same or change as per requirements.

All the metadata of the documents will be created and maintained. Records for the configuration management will be consistently maintained. All the changes in the documents will be recorded. Dates of the release of documents and their new versions will be recorded. Dates of approval for changes in documents will be recorded. Dates of changes in documents will be recorded.

## 7.2 Independent Verification and Validation

Unit Testing will be performed for validation purposes. Team Members will review modules completed by other members for verification. If bug is found in some component, it will be retested after removing the bug. Admin will review the working of all the components and make sure that they will be meeting criteria for QA.

## 7.3 Documentation

Documents will be reviewed by the team members working in specific area. Front End, Back-end and iOS developers will review the documents related to their area. Documents will be created using some standard format. Proper guidelines will be followed while creating the documents. Documents will be released before or on fixed dates. All the team members will be able to access all the documents.

Additional details TBD.

Table 7-1. Statusphere Documentation

| **Document Type** | **Format Standard** | **Estimated Page Count** | **Peer Review Type** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

## 7.4 Quality Assurance

To develop high quality software, we must ensure that it meets the requirements listed by the customer. A Software Quality Assurance plan with the following specifications should be adopted:

* Software error must be reported. Since GitHub will be used, this will be easy to implement. We must then determine whether the error was due to incomplete or erroneous specifications.
* Once the cause of the error has been determined, measures should be taken to correct them.
* The software will be reviewed with the customer every two weeks to avoid any miscommunication of requirements.
* The intuitiveness of the user interface must be determined by conducting reviews with the customer and test users (to be provided by the customer).
* Effective testing of the software should be conducted to ensure its error free functioning.

## 7.5 Reviews and Audits

Most the project reviews will be conducted by our peers and team leader during weekly meetings. It is here that the issues with development will be expressed and defined, as well as corrective actions to be taken. We will meet the customer every two weeks to ensure that the project meets the requirements and to update the customer on the development progress.

## 7.6 Problem Resolution

Depending on the scale of the problem, the resolution may occur at an individual level or at the team level. The problems that impact the project will need to be discussed with the entire team during the weekly meetings and a resolution will need to be arrived at collectively; root cause analysis will be applied, when appropriate, to ensure that problems are correctly identified. Interpersonal problems will be resolved in the presence of a mediator, either the project manager and/or the project coordinator (Dr. Edwin Nassiff), whichever is the most appropriate.

Additional details TBD.

## 7.7 Contractor Management

The project does not require any work by external contractors.

### 7.7.1 Contracting Process

N/A

### 7.7.2 Contractor Performance Monitoring

N/A

## 7.8 Process Improvement

TBD.

The following paragraphs provide data on the Statusphere efforts for continuing process improvement.

### 7.8.1 Systems/Software Process Improvement Lead

TBD

### 7.8.2 Systems Engineering Process Group

TBD.

# SECTION 8. Additional Plans

***IEEE Std 1058-1998 Guidance***

***(Clause 8) Additional plans***

*This clause shall contain additional plans, or activities, required to satisfy product requirements and contractual terms.*

*Additional plans for a particular project may include plans for assuring that safety, privacy, and security requirements for the product are met, special facilities or equipment, product installation plans, user training plans, integration plans, data conversion plans, system transition plans, product maintenance plans, logistic engineering approach, or product support plans.*

**APPENDICES**

***IEEE Std 1058-1998 Guidance***

*Annexes may be included, either directly or by reference to other documents, to provide supporting details that could detract from the document if included in the body.*

***General Guidance***

###### In this template, the following appendices are used for reference purposes only. It should not be assumed that the referenced documents exist as an example.

# Appendix A. StatusphereMaster Schedule (Microsoft Project)

***Guidance***

*The objective of the Statusphere master schedule is to provide management with the task map and tracking tool needed to guide the project in the performance of its mission.*

The Statusphere *master schedule’s Microsoft Project representation of the WBS would be tailored from the templates available from the SSC Pacific Process Asset Library (PAL) in the “SW-CMM Archive”. Draft Microsoft Project templates are found under the “Process Assets by SW-CMM KPA”, “Software Project Planning (SPP)” in the “Tools” section. These templates can be tailored up or down to meet specific project needs.*

# Appendix B. StatusphereFacilities Plan

***Guidance***

*The objective of the Statusphere Facilities Plan is to document the environmental needs of the project. These needs include space, equipments, security, safety, support tools, and the staff necessary to maintain and operate an environment needed for project operations.*

The facilities requirements for projects vary broadly, often with several projects sharing both facilities and computer resources. There currently are no templates available from the SSC Pacific PAL to assist in developing a facilities plan. However, recommended issues to address in a Facilities Plan would include, but not be limited to, the following list:

*1. Facility Objectives/General Description*

*2. Facility Locations (i.e., Building Locations)*

*3. Facility Diagrams*

*a. Floor Plans (i.e., lab, work cubicles)*

*b. Environmental Requirements i.e. Heating, Lighting*

*4. Facilities Equipment Requirements*

*a. Equipment Lists (i.e., work stations, development, test)*

*b. Equipment Interface Diagrams*

*c. Space Equipment Layouts*

*d. Inspections and Records Requirements*

*5. Facilities Software Requirements*

*a. Software by Development/Test Host Equipment*

*b. Software by Workstation*

*6. Facilities Operating Personnel Requirements*

*7. Facilities Operating Personnel Training Requirements*

*8. Security Measures*

*a. Internal*

*b. External*

*9. Safety Measures*

*10. Maintenance Requirements (i.e., spaces, per equipment)*

*11. Facilities Performance Measurements*

# Appendix C. StatusphereProject Training Plan

***Guidance***

*The objective of the Statusphere Project Training Plan is to develop the skills and knowledge of the project staff so they can perform their roles effectively and efficiently.*

*The Statusphere Project Training Plan would be tailored from the* [*Department/Project Training Plan Template*](http://sepo.spawar.navy.mil/Department_Project_Training_Plan_Template.doc) *available from the SSC Pacific Process Asset Library (PAL). The template is located in the “Process Assets” sub-page under the “Organizational Training” PA in the “Plans” section. The template can be tailored up or down to meet specific project needs.*

# Appendix D. Statusphere Measurement Plan

***Guidance***

*The objective of the Statusphere Measurement Plan is to develop and present the data needed to support project management information needs necessary to ensure objective decision-making.*

*The Statusphere Measurement Plan would be tailored from the Software Measurement Plan Template available from the SSC Pacific Process Asset Library (PAL) in the “SW-CMM Archive”. This template can be found under the ”Process Assets by SW-CMM KPA”, “Software Project Tracking and Oversight (SPTO)” KPA in the “Tools” section. The template can be tailored up or down to meet specific project needs.*

# Appendix E. Statusphere Product Engineering and Qualification Process

***Guidance***

*The objective of the Statusphere Product Engineering and Qualification (PE&Q) Process is to document the processes comprising a technical solution for development, maintenance, test, and product qualification.*

*The Statusphere PE&Q Process would be tailored from the Product Engineering and Qualification Process available from the SSC Pacific Process Asset Library (PAL). The process is located in the “Process Assets” sub-page under the “Technical Solution” PA in the “Process” section. The PE&Q Process can be tailored up or down to meet specific project needs.*

# Appendix F. StatusphereQuality Assurance Plan

***Guidance***

*The objective of the Statusphere Quality Assurance Plan is to provide staff and management with objective insights into processes and associated work products, ensuring their conformance to documented requirements.*

*The Statusphere Quality Assurance Plan would be tailored from the Quality Assurance Plan Template available from the SSC Pacific Process Asset Library (PAL). The process is located in the “Process Assets” sub-page under the “Process and Product Quality Assurance (PPQA)” PA in the “Plans” section. The template can be tailored up or down to meet specific project needs.*

# Appendix G. StatusphereConfiguration Management Plan

***Guidance***

*The objective of the Statusphere Configuration Management Plan is to establish and maintain the integrity of Statusphere work products using configuration identification, configuration control, configuration status accounting, and configuration audits.*

*The Statusphere Configuration Management Plan would be tailored from the Configuration Management Plan Template available from the SSC Pacific Process Asset Library (PAL). The template is located in the “Process Assets” sub-page under the “Configuration Management (CM)” PA in the “Plans” section. The template can be tailored up or down to meet specific project needs.*

DOCUMENT CHANGE REQUEST (DCR)

|  |  |
| --- | --- |
| Document Title: **[[Project]]****Project Management Plan** | Tracking Number: |
| Name of Submitting Organization: | |
| Organization Contact: | Phone: |
| Mailing Address: | |
| DCR Description: | Date: |
| Change Location:  (use section #, figure #, table #, etc.) | |
| Proposed change: | |
| Rationale for Change: | |
| Note: For the ***appropriate authority*** to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale.  Send to: Commanding Officer, Space and Naval Warfare Systems Center, Code [[xxx]], 53560 Hull Street, San Diego, CA 92152-5001  Fax to: ***indicate appropriate fax number***  Email to: ***indicate appropriate email***  Submit online: ***indicate appropriate URL***  DCR Form 1/2009 | |