**National Aeronautics and Space Administration**

**(NASA) Extra-vehicular Activity (EVA) Phase 4**

**Project Management Plan**

Tristan Arjoon, Mario Curtis, Shane Farmer, Claudel Guembu

Sean Johnson,

SWEN 670 9040 (2188)

UMUC Graduate School of Management and Technology

Software Engineering Project

30 September 2018

Fall Semester

**Table of Contents**

[1. Introduction 5](#_Toc527913953)

[1.1 Purpose 5](#_Toc527913954)

[1.2 Project Team 5](#_Toc527913955)

[1.3 Scope 5](#_Toc527913956)

[1.4 Goals and Objectives 6](#_Toc527913957)

[1.5 Overview 6](#_Toc527913958)

[2. Detailed Objectives (Milestones) 6](#_Toc527913959)

[3. Statement of Work (SOW) 8](#_Toc527913960)

[4. Assumptions and Constraints 9](#_Toc527913961)

[4.1 Assumptions 9](#_Toc527913962)

[4.2 Constraints 10](#_Toc527913963)

[5. Work Breakdown Structure (WBS) 11](#_Toc527913964)

[Work Breakdown Structure (Dates) 11](#_Toc527913965)

[Work Breakdown Structure (Tasks) 12](#_Toc527913966)

[6. Possible User Interfaces 13](#_Toc527913967)

[6.1 User interface requirements 13](#_Toc527913968)

[7. Software Requirement Specification (SRS) 14](#_Toc527913969)

[7.1 Product Functions 14](#_Toc527913970)

[7.2 Hardware Interfaces 14](#_Toc527913971)

[7.4 Software Interfaces 14](#_Toc527913972)

[8. Software Development and Implementation 15](#_Toc527913973)

[8.1 Software Development Method 15](#_Toc527913974)

[8.1.1 Background 15](#_Toc527913975)

[8.1.2 Agile Methodology 15](#_Toc527913976)

[8.1.3 Programming Languages 15](#_Toc527913977)

[8.1.4 Development Environment 15](#_Toc527913978)

[8.1.5 Source Control 16](#_Toc527913979)

[9. Test Plan 16](#_Toc527913980)

[10. References 17](#_Toc527913981)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Date | Description |
| 1.0 | Mario Curtis | 9/18/18 | Created document. |
| 1.1 | Tristan Arjoon | 9/18/18 | Updated Draft Content |
| 1.2 | Sean Johnson | 9/25/18 | Added Software Development and Implementation section |
| 1.3 | Tristan Arjoon | 9/25/18 | Added Detailed Objectives section and edits. |
| 1.4 | Ryan Murphy | 9/29/18 | Added Assumptions and Constraints |
| 1.5 | Sean Johnson | 9/29/18 | Added Test Plan section |
| 1.6 | Mario Curtis | 9/30/18 | Added the WBS from Project file |
| 1.7 | Shane Farmer | 9/30/18 | Added Statement of Work section |
| 1.8 | Tristan Arjoon | 9/30/18 | Update overall Draft Content (all sections) |
| 1.9 | Tristan Arjoon  Shane Farmer | 9/30/18 | Updated SoW |
| 2.0 | Tristan Arjoon | 10/15/18 | Update to V2 PMP & Data |
| 2.1 | Mario Curtis | 10/21/18 | Update to PMP WBS |

**ii. Contributions**

This section is to acknowledge the contributions made by previous groups towards the completion of the NASA Path Project.

Phase 1 Group Members

* T. Hill
* E. Guthrie
* J. Dovan
* D. Stine
* A. Akinyele
* P. Kyaw

Phase 2 Group Members

* Nicoleta Florea
* Bradley Hankinson
* Brittany Ofori
* George Wheeler
* Justin Widmann

Phase 3 Group Members

* Fritz Cenal
* Jennifer Dostal
* Cameron Farley
* Lincoln Powell
* Deepali Varma
* Tenadam Weldesemayat

## 

## 

## 

# 1. Introduction

## 1.1 Purpose

The NASA Path project is intended to be an aid in training simulations and real-world Extra Vehicular Activity (EVA or “space walks”). The purpose of this document is to improve The NASA EVA Path Finder software for all personnel to connect the best way possible, so each member could use it effectively.

The application utilizes 3D modelling (Dynamic Onboard Ubiquitous Graphics or DOUG) to include all the modules and hardware on the International Space Station (ISS), as well as each handrail.

The aim of the presumed application is to allow a user to input start and end points (handrails) along with route options such as hazard avoidance, user wingspan and safety tether routing.

This project plan outlines the planning considerations and strategic approach taken by Phase 4 of the NASA EVA Path project. This document is intended for all project team members who will (1) direct development team priorities, (2) implement desired features or (3) verify the correct functioning of the system.

## 1.2 Project Team

* Tristan Arjoon
* Mario Curtis
* Shane Farmer
* Claudel Guembu
* Sean Johnson
* Ryan Murphy

## 1.3 Scope

This project plan provides an introductory direction for the NASA EVA Path Phase 4 project team. The aim for this plan is to be evolutionary -- to be updated as necessary to fit the current demands of the project and as priorities shift. Thus, this plan in its current state is a current reflection of the strategic approach made by the project team. Topical areas within will adjust as deemed appropriate and necessary. The NASA EVA Path Finder software is an effort to simulate a list of concepts that don’t exist in order to write a software for the system using different tools to lead the programming team to produce it.

Overall, this PMP outlines the strategic lifecycle approach for delivering the customer critical documents and source code that improves upon the latest phase of EVA.

## 1.4 Goals and Objectives

Our project objective is to generate maps and navigation for the exterior of the ISS using handrails and structural beams via DOUG (Dynamic Onboard Ubiquitous Graphics) which is a 3D model of the exterior of the Space Station, complete with handrails and other interfaces useful to a space walker. The NASA EVA Path Finder software will calculate and display:

* Pathing algorithm for shortest distance.
* Hazard counter on all relevant paths.
* Longer arm reach.

Phase 4 will enhance the above features already implemented and focus on the below requirements outlined by the customer:

* Expand the code to make use of the entire ISS model.
* Initiate and produce an estimated 3 functional requirements outlined in the customer requirement document. (This will be updated when team members iron out limitations)

## 1.5 Overview

Below if an outline of the current Project Management Plan.

* Section 1 of this document provides purpose, scope, goals and objectives.
* Section 2 gives a more detailed objectives of the document.
* Section 3 contains the statement of work of the project.
* Section 4 details all the assumptions and constraints of the document.
* Section 5 contains the work breakdown structure (WBS).
* Section 6 discusses possible User Interfaces (UIs).
* Section 7 contains Software Requirement Specification (SRS).
* Section 8 contains the Software Development and Implementation.
* Section 9 contains the Test Plan.
* Section 10 contains the References.

# 2. Detailed Objectives (Milestones)

Nasa representative highly requested that we followed the direction of Phase 3’s delivery packages. A deep dive of Phase 3 lessons learned, and product deliverables will be taken into account when building Phase 4 of EVA. To keep the customer satisfied, Phase 4 will adhere to an agile based methodology which incorporates sprints. These sprints will happen at the end of each of the four milestones; the milestones take place every 3 weeks. See section 1.4 for

Below are the milestones which include package deliverables:

**Milestone 1: Week 3 (September 30th)**

* Project Management Plan (baseline)
* Presentation

During milestone 1, the team actively communicates to iron out expectations of the customer. As well as getting acclimated with previous phases source code. A baseline project plan will be created based on customer expectations and strategic approach. The Project Plan will be a revolving document that can be altered based on stakeholder feedback (customer/professor). In addition, a presentation will be given to stakeholders.

**Milestone 2: Week 6 (October 21)**

* Project Management Plan (updated)
* Feature documentation and Use Case Testing
* Working code (in repository)
* Working Design Documentation
* Presentation (update)

This milestone will deliver an updated PMP and feature document. The feature document outlines incorporated code and a use case test plan. In addition, any updated versions of working code will be uploaded to the repository.

**Milestone 3: Week 9 (November 11)**

* Project Management Plan (updated)
* Feature documentation and Use Case Testing
* Working code (in repository)
* Working Design Documentation
* Presentation (update)

This milestone will deliver an updated PMP and feature document. The feature document outlines incorporated code and a use case test plan. In addition, any updated versions of working code will be uploaded to the repository.

**Milestone 4: Week 12 (December 2)**

* Project Management Plan (updated)
* Feature documentation and Use Case Testing
* Finalize Working code (in repository)
* Finalize Design Documentation
* Presentation (final)

This milestone will deliver an updated PMP and feature document. The feature document outlines incorporated code and a use case test plan. In addition, any updated versions of working code will be uploaded to the repository.

# 3. Statement of Work (SOW)

**Scope of work**

Phase 4 will work in a team-based Agile environment. NASA, the client, has provided previous development teams with high-level requirements--the resulting backlogs will be the basis for the current development team’s work. The client and development team will work together closely to clarify user stories and establish criteria for acceptance.

The goal for this statement of work (SOW) is to organize as best as possible the work required for successful completion of backlog items agreed upon with the client. The scope includes, but is not limited to the following:

* **Scope of work**
* **Project objectives**
* **Project schedule**
* **Deliverables**
* **Expected outcomes**

**Project Objectives**

(See 1.4 of Project Plan)

The primary objective of this project is to provide the client additional desired functionality of the NASA EVA Path Finder software. Though currently functional, there are a number of backlog items that are desired, 4 of which will be addressed by this development team (subject to change, depending on depth, constraints, understanding, etc).

* Add indications of when axial direction and plane changes occur
  + e.g. from port to zenith or from face 1 to face 2
* Provide a legend/key on how to maneuver ISS
* Expand the code to make use of the entire ISS model
  + This is the highest priority item per the client

**Project Schedule**

(See Section 5 of Project Plan)

This project will be completed in 4 milestones, 3 weeks apart. The specifics within each milestone will be updated as the project progresses.

* Milestone 1: Week 3 (September 30th)
  + Project Management Plan
  + Presentation (Oct 1st at 6:30PM)
* Milestone 2: Week 6 (October 21st)
* Milestone 3: Week 9 (November 11th)
* Milestone 4: Week 12 (December 2nd)

**Deliverables**

* Create a baselining effort. Implement proposed methodology through 4 milestones.
* The development team will develop, configure, test, and submit iterations by applying an iterative process utilizing the proposed Agile methodology and agreed upon release cycle
* Deliverables shall be provided on the milestone dates specified; any changes will require approval from the professor.
* The development team will wholly review all deliverables prior to submission.

**Expected Outcomes**

The expected outcome of each subsequent milestone is working software addressing desired functionality by the client. As the project matures, we will have a greater grasp of the specifics of each milestone. At a minimum, we will address the 4 project objectives stated above. All in all, we aim to improve functionality of the NASA EVA Path Finder software.

# 4. Assumptions and Constraints

Based off the requirements that have been acquired from the stakeholder as well as knowledge of the basic format of the class, the team is moving forward with several assumptions about the nature of the project. Also based off this knowledge, several constraints have been identified that will impact the projects development. There are listed now below.

## 4.1 Assumptions

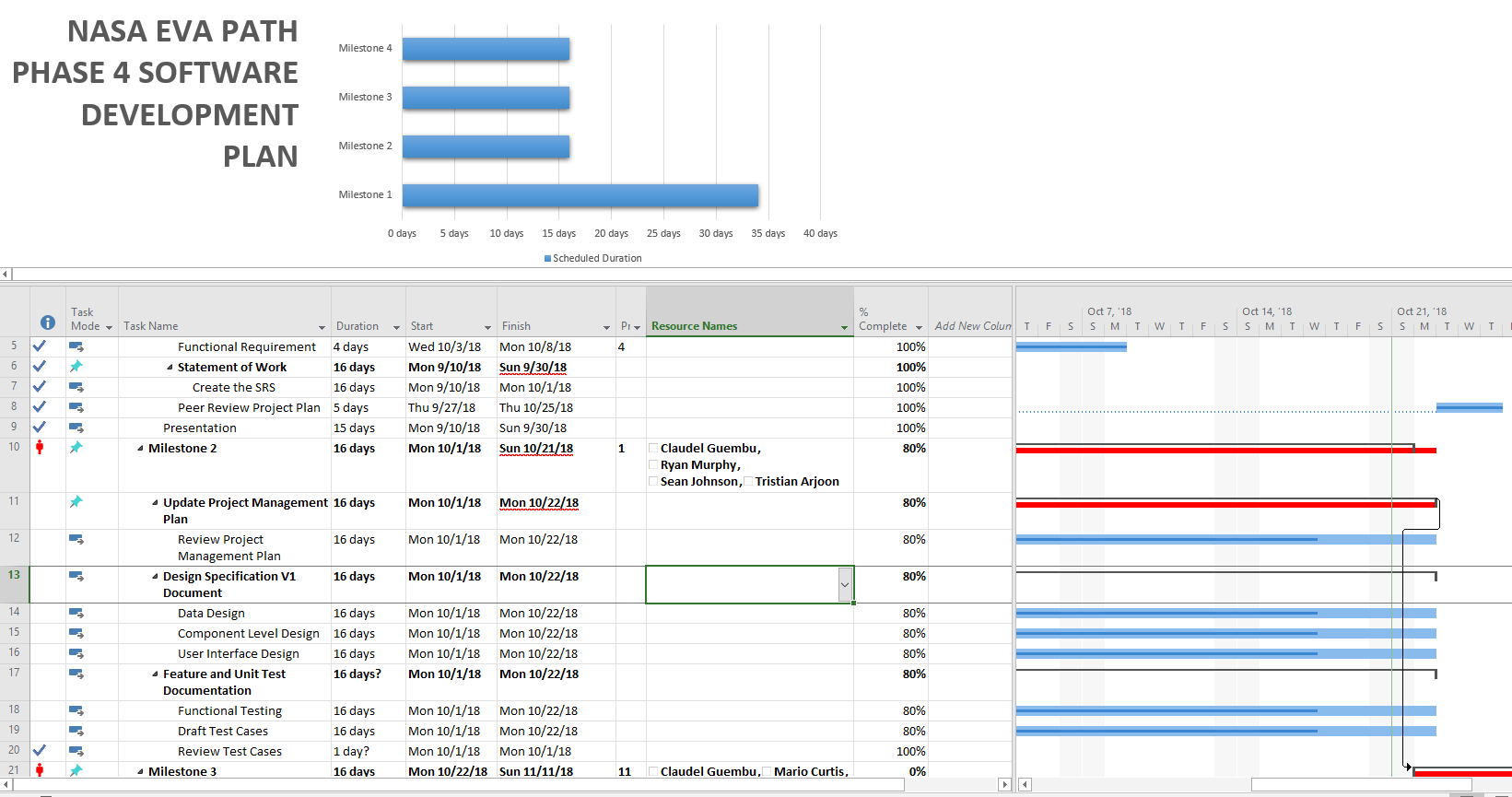
* The GitHub repository provided by the stakeholder is the most recent, up-to-date version of the codebase and the code is all housed in one location (i.e. there are no other places the team will need to visit to get all the code).
* The purpose of this project is to expand features and resolve defects from an existing application and not perform other tasks, such as rewriting the code base to another language
* The existing technology stack used in the project will remain sufficient for the stakeholder’s requirements and will not require migration to other technologies. Note that this does not exclude the possibility of additional frameworks or technologies being added.
* The technology stack, as well as all development tools and support applications are available to all members of the team and will be able to use them effectively.

## 4.2 Constraints

* The length of this project is dictated by the course it is attached to, SWEN 670 (2188). SWEN 670 is a 12-week course, ending on December 02, 2018. At the time the course will be completed and all work on the project must be completed as well.
* Team members are working across several time zones, potentially making it challenging to sync up.
* The team's ability to respond to last minute changes is constrained by each individual member limited schedule. As each team member has additional commitments outside of class (such as work, other classes, etc.) it seems likely that the team would struggle if they were presented with a sudden change with a short suspense.
* As the project’s deliverables are broken up into four set milestones with set dates for each, the work for each milestone will need to be divided up beforehand.

# 5. Work Breakdown Structure (WBS)

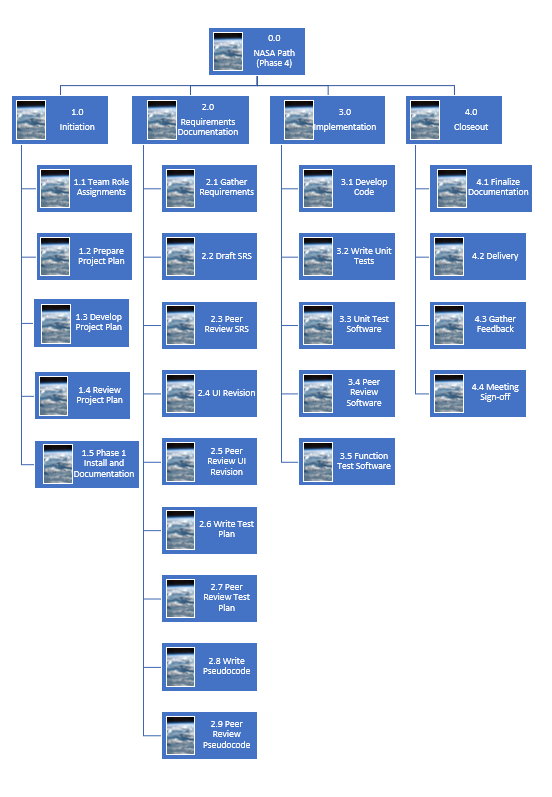
## Work Breakdown Structure (Dates)

****

# 

## Work Breakdown Structure (Tasks)

The below WBS shows a waterfall approach, but this approach will be done in cycles from Requirements Documentation to Closeout for each milestone.

****

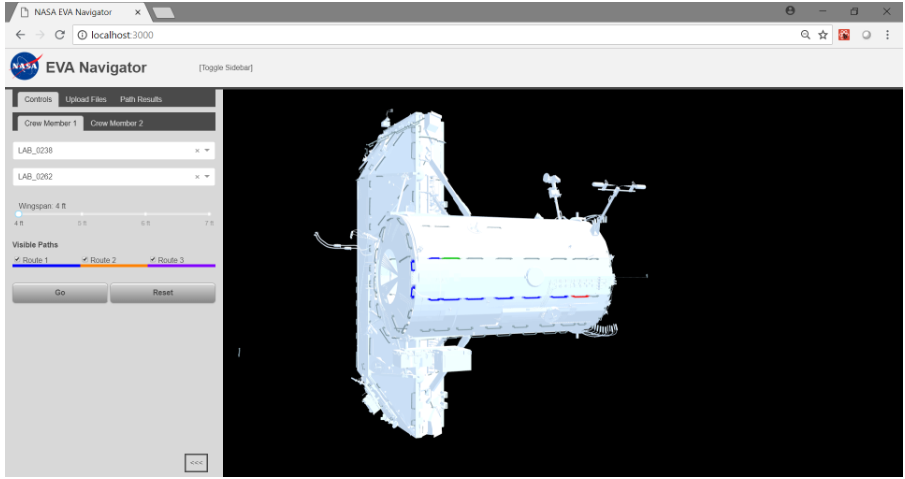
# 6. Possible User Interfaces

## 6.1 User interface requirements

According to the stakeholder’s initial requirements, the user interface should allow users to minimally:

* Select start and end points, each different handrail.
* Choose options for the route.
  + Default will be the shortest route.
  + Route that calculates the least number of hazards (e.g. sharp edges, radiating hardware, articulating structures or shatter-able materials).
  + Route which de-conflicts their partner’s route (EVAs are performed by a pair of crews).
  + Route including one or more waypoints in the middle of the route.
  + Route with the fewest number of rotations and plane changes.
* Choose options for user wingspan (this is important in places where handrails are spaced farther apart than some people’s reach limit).
* Choose options for safety tether routing.

Here is a snapshot of Phase 3 current interface:



This phase (phase 4) will focus on:

* Updated model to include all of ISS. (Milestone 2 Complete)
* Provide a legend/key on how to maneuver ISS.
* Add indications of when axial directions and plane changes.
* Display tether routing (If ahead of schedule).

Additionally, the phase one, two, and three software could only display individual sections of the ISS while phase four requires that the entirety of the ISS be displayed as a 3D model. This will require changes to code for displaying an interactive, full model of the station and its paths.

# 7. Software Requirement Specification (SRS)

## 7.1 Product Functions

The primary functionality of this software is to perform calculations of the shortest, optimal paths for astronauts during EVA, or “spacewalks” and display them using a 3D model of the entire International Space Station (ISS). The software is to allow the user to pick “start” and “end” points using dropdown controls or by clicking the handrails on the displayed model of the ISS and the software will then perform the necessary calculations. When determining the optimal path, the software will also provide interfaces for other factors to be considered when making the calculations. These additional factors include determining optimal path based on wingspan (i.e. distance between handrails), determining paths for two crew members that de–conflicts their potential routes, and determining optimal path for a route with one or more waypoints. Other factors that are to be considered that are not entered by the user into the interface are determining paths by minimizing rotations and plane changes and displaying and de–conflicting tether routes for crew members.

## 7.2 Hardware Interfaces

Hardware specifications for the operating environment have remained unchanged since phase one of the NASA Path project. These specifications include at a minimum:

* Monitor/Keyboard/Mouse support
* An internet connection (TCP/IP) to download/upload files
* The client computer must have at least 8GB of random-access memory (RAM)
* The client computer must have at least 1GB for the graphics processing unit (GPU)
* The client computer must have a 3.4 GHz central processing unit or faster
* The client computer must have at least 10GB of free disk space

## 7.4 Software Interfaces

Software interface requirements for the web browser and operating system have remained unchanged since phase one of the NASA Path project. These specifications include:

* The client computer must have Chrome, Firefox, or IE 11 web browsers installed
* The client computer must be running Windows 7 (or higher), Mac OS X (version 10.7.x or higher), or Ubuntu 14 or higher.

To run the current and any future builds of the software, the following software interfaces are needed at minimum:

* Java 8 (development platform)
* Yarn (package manager)

# 8. Software Development and Implementation

## 8.1 Software Development Method

### 8.1.1 Background

The different phases or teams of the NASA EVA Path project had to find a software development methodology that would work depending on each team’s demands and needs. The software development methodology is defined as a framework that is used to structure, plan and control the process of developing an information system (Software Development Methodologies, n.d.). Phase 1 of the project used the Scrum methodology. Phase 2 picked up using the feature driven development methodology. Phase 3 used the Agile methodology. For Phase 4, the decision was made to use the Agile methodology as well

### 8.1.2 Agile Methodology

As stated above, the agile methodology will be used for Phase 4 software development. This methodology was picked because the team will develop the software in short iterations called sprints. Our sprints will last between two to three weeks. At the end of each sprint, a milestone will be delivered. Also, at the end of each sprint, there will be a lesson learned where the team evaluates what worked and what didn’t work so that the team will become efficient. The agile methodology emphasizes working software to measure progress.

### 8.1.3 Programming Languages

The programming languages that are used in the NASA EVA Path are Java and JavaScript. Scripting languages such as Yarn and Docker have been used for deploying the application as well

### 8.1.4 Development Environment

The development environment that will be used for the project is dependent on Docker. Docker is a container platform which allows an application and its dependents to be encapsulated in a container. Using a container platform will allow the application to run alone without depending on systems to be configured or installed for the application.

### 8.1.5 Source Control

The NASA EVA Path project will use Git as the version control system and GitHub as the code repository to meet the requirement of the final software being readily available as free and open source code.

The Phase 4 project will duplicate Phase 3’s GitHub project repository <https://github.com/xpaddict/nasa-path-finder>.

An initial pull request will be made from Phase 4’s duplicated repository to the customer’s initial project repository, <https://github.com/darenwelsh/EVANav>, to update the duplicate repository to Phase 3’s final. The additions and revisions will then be made in the duplicate repository. When the software is in a good working state, pull requests will be made to the customer’s project repository.

# 9. Test Plan

The test plan for Phase 4 includes the performance of unit and integration testing. Each feature will be unit tested using JUnit tests with test cases that will provided by the customer. Once the features/functions/methods are deemed to have passed unit testing, they will then run through integration testing. There will also be regression testing performed to make sure that when a new feature is introduced to the system, that the new feature didn’t break existing code. When the testing is completed, the features will be merged to a pre-production state and be deemed ready for prototyping.

# 10. References

* Baars, Wouter (2006). “Project Management Handbook”. DANS – Data Archiving and Networked Services. ISBN 90 6984 496 6. Retrieved from: ‘Week 1 folder’
* Software Development Methodologies. (n.d.). Retrieved from <http://www.itinfo.am/eng/software-development-methodologies/>
* Project Management Institute. (2013). Project Risk Management. In P. M. Institute, *A Guide to the Project Management Body of Knowledge.* Newtown Square: Project Management Institute, Inc.