**Software Project Management Plan**

**Schedule Assistant**

2.02.15

**Team Members**

Michael Harris William Freeman

Jonathan Davis Samaa Gazzaz

Raphael Fontes de Silva Adam Guerrero

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| 1 | 1/25/2015 | Initial release |
| 2 | 1/25/2015 | Revision |
| 3 | 2/2/2015 | Revision |
| 4 | 2/12/2015 | Revision |
| 5 | 2/20/2015 | Revision |
| 6 | 2/22/2015 | Revision |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Document Storage**

This document is stored in the GitHub repository at: \Spring2015Team3\Documentation

**Document Owner**

Adam Guerrero is responsible for developing and maintaining this document.**Table of Contents**

[1 Overview 4](#_Toc199052960)

[1.1 Purpose and Scope 4](#_Toc199052961)

[1.2 Goals and Objectives 4](#_Toc199052962)

[1.3 Project Deliverables 5](#_Toc199052963)

[1.4 Assumptions and Constraints 5](#_Toc199052964)

[1.5 Schedule and Budget Summary 6](#_Toc199052965)

[1.6 Success Criteria 6](#_Toc199052966)

[1.7 Definitions 6](#_Toc199052967)

[1.8 Evolution of the Project Plan 6](#_Toc199052968)

[2 Startup Plan 6](#_Toc199052969)

[2.1 Team Organization 6](#_Toc199052970)

[2.2 Project Communications 7](#_Toc199052971)

[2.3 Technical Process 7](#_Toc199052972)

[2.4 Tools 7](#_Toc199052973)

[3 Work Plan 7](#_Toc199052974)

[3.1 Activities and Tasks 7](#_Toc199052975)

[3.2 Release Plan 8](#_Toc199052976)

[3.3 Iteration Plans 8](#_Toc199052977)

[3.4 Budget 8](#_Toc199052978)

[4 Control Plan 8](#_Toc199052979)

[4.1 Monitoring and Control 8](#_Toc199052980)

[4.2 Metrics Collection 8](#_Toc199052981)

[5 Supporting Process Plans 8](#_Toc199052982)

[5.1 Risk Management Plan 8](#_Toc199052983)

[5.2 Configuration Management Plan 8](#_Toc199052984)

[5.3 Verification and Validation Plan 9](#_Toc199052985)

# Overview

## Purpose and Scope

The On-Call Assistant will create an equitable schedule of on-call rotations for team members based on the following considerations:

* When team members have requested out-of-office time
* Previous rotations for which team members have been scheduled
* Previous rotations that include paid holidays for which team members have been scheduled

The system, intended for a single user, will operate in a browser hosted locally on the user’s machine. As such, it will not provide security or user authentication measures.

## Goals and Objectives

1. Provide a usable and efficient system that allows managers to view current and past schedules and to create dependable future schedules.
2. Create reliable schedules that minimize the effort that, in the past, has been required to manually create and update a schedule of on-call rotations. In return this should increase workflow and decrease confusion.
3. The On-Call Assistant will offer a simplistic and user friendly interface. The user will not have to worry about managing complex back-end storage and development. Should the user require more complex functionality, such functionality will be available with configuration files and an open source delivery.

## Project Deliverables

1. Vision Statement
2. Project Charter
3. Software Requirements Specification (SRS)
4. Software Project Management Plan (SPMP)
5. Release Plan
6. Iteration Plan
7. Memo of Understanding
8. Project Success Criteria
9. Project Closure Report
10. Change Request
11. Change Control Log
12. Issue Log
13. Status Report
14. Architecture and Design Documentation
15. Coding Standards
16. Test Plan
17. Test Case Specification
18. User Guide

## Assumptions and Constraints

1. It is assumed that the application is intended for a single user. Access control will not be provided in the software.
2. It is assumed that the application will be hosted locally on the user’s computer with no internet exposure. As such, no authentication or security measures will be included in the initial release.
3. It is assumed that a quarter is a ‘fiscal’ quarter. By default, each schedule of on-call rotations will span a fiscal quarter.
4. It is assumed that the user will be available to make minor adjustments to a schedule should a team member’s availability change after said schedule has been generated. In the event that such a change should require more than minor adjustments, the user will have the ability to regenerate the schedule within a time span of his/her specification.

## Schedule and Budget Summary

See project timeline file, listed as Project Timeline under Documents folder

## Success Criteria

* Projected hourly requirements does not exceed estimate in charter (420 hours) by more than 20%
* All user stories from product backlog up through priority 4 must be completed by May 10th
* User interface is accepted by customer during a presentation before May 10th deadline

## Definitions

This section should define potentially unfamiliar or ambiguous words, acronyms and abbreviations. **TBD**

## Evolution of the Project Plan

Prior to each iteration the team will have a formal meeting to discuss next steps in the project. The meeting will encompass project updates, and an agenda created by the team leader. Key take away from weekly meetings are each team member should distinctly know their tasks for the next iteration as well as their time-line for said task. The development branch will be updated by team members as work is completed. Currently there is a branch off of GitHub that is primarily for any updates and edits before finalizing. The team leader is solely responsible for finalizing and merging development changes into the master branch for delivery upon completion of each iteration.

Risk mitigation efforts will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

# Startup Plan

## Team Organization

**Project Manager: Michael Harris**

Achieves operational objectives by contributing information and recommendations to strategic plans and reviews. Will prepare and complete action plans; implementing production, quality, and customer-service standards. Will also be responsible for resolving problems; determining application improvements; implementing change, and maintaining communication between the client and the organization.

**Front End developer: Samaa Gaaza**

In charge of UI design and the aesthetics of the website. Will be responsible for the layout and design of the system application webpage.

**Back end developer: William Freeman, Raphael Fontes da Silva**

Primary focus is to manage the interchange of data between the application and the browser. Will focus on the server-side of programming for the web, will also understand the programming languages which control the display of content in the browser. (See “Front End developer”)

**Database Administrator: Johnathan Davis:**

Will identify database requirements by interviewing customers; analyzing department applications, programming, and operations; evaluating existing systems and designing proposed systems.

Will also be responsible for recommending solutions by defining database physical structure and functional capabilities, database security, data back-up, and recovery specifications.

**Administration: Adam Guerrero**

Responsible for written and oral communication, word processing and typing, and organizational and presentation skills, as well as the ability to multi-task and work well under pressure.

## Project Communications

* A group has been created in the UMKC Outlook server for distributing formal communication to the group.
* A Trello organization has been created for managing tasks and updating team members with changes or for requesting information.
* A weekly meeting occurs every Friday to discuss difficulties and plan iterations.

## Technical Process

For this project the team took an iterative and incremental approach. Utilizing the [agile software development](http://en.wikipedia.org/wiki/Agile_software_development) framework for managing product development. It defines "a flexible, [holistic](http://en.wikipedia.org/wiki/Holism) product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines in the project.

## Tools

* Programming Language –C#
* Defect tracking – defects and issues will be tracked using Trello
* Build tools – local and main builds will be done using Visual Studio.
* Automated testing – unit tests will be implemented with the CUnit testing framework.
* Github social repository.

# Work Plan

## Release Plan

See Project Timeline document under Documentation folder for planned release timeline. At the end of each iteration, a stable release will be merged into the master branch.

## Iteration Plans

See Project Timeline document under Documentation folder for planned actions on iterations.

# Control Plan

## Monitoring and Control

Project manager will perform merges into master branch after running tests to ensure stability of project prior to release. Each iteration concludes with a review. Iterations are planned in Project Timeline document.

## Project Measurements

|  |  |  |
| --- | --- | --- |
| **Phase** | **Measurement** | **Source** |
| Release Planning | Record effort estimates for product features | Mgr |
| Iteration Planning | Record effort estimates for scheduled tasks  Update effort estimates for product features  Update estimated dates in release plan | Mgr |
| Iteration Closeout | Record actual effort for scheduled tasks  Record actual effort for product features  Record LOC count for modules written | Mgr/Pgr |
| System Test | Record the rate at which errors are found. | QA |
| Project Closeout | Archive project performance data in process database. (See process database definition for a list of measures to record.) | Mgr |
| Ongoing | Record defects found from integration testing through first year of release.  Assign each defect to one of the following categories: blocker, critical, major, minor or trivial. Keep track of the state of each defect: open, assigned, fixed, closed. | Mgr/Pgr/QA |

# Supporting Process Plans

## Risk Management Plan

Risk 1. Software development, given the intangible nature and uniqueness of software, is inherently difficult to estimate and schedule.

Risk 2. As the project progresses more and more features that were not identified at the beginning of the project emerge that threaten estimates and timelines.

The project manager has led the project team in developing responses to each identified risk. As more risks are identified, they will be qualified and the team will develop avoidance and mitigation strategies. These risks will also be added to the Risk Register and the Project Plan to ensure they are monitored at the appropriate times and are responded to accordingly. If necessary, the Risk Management Plan will be updated.

The risks for this project will be managed and controlled within the constraints of time, scope, and cost. All identified risks will be evaluated in order to determine how they affect this triple constraint. The project manager, with the assistance of the project team, will determine the best way to respond to each risk to ensure compliance with these constraints.

|  |  |
| --- | --- |
| Limited experience with C# and [ASP.NET](http://ASP.NET) leading to difficulty in estimating timelines. | We reduced this risk by buying information with our technical prototype. |
| Limited experience with web technologies which could make design challenging. | Risk mitigation – Setting deadlines allows to track progress and ensure we are on track to deliver system requirements at the end of each iteration. Thus mitigating the risk. |
| Limited experience with distributed version control software | Risk acceptance –This is software is mandatory for the project. The team will gain experience as the project continues. The risk will gradually decrease as experience with the software increases. |
| Narrow scope of project. | In order to get detailed instructions, and specifics for system and system requirements the team has access to the client anytime we need clarification. |
| Web based tools don't work as expected. | Through web based tutorials and other learning tools we reduce this risk. This risk poses problems because we have to learn a whole new software which makes it difficult to estimate timelines. |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Risk | Risk Response |

## Configuration Management Plan

Configuration management plans for this document and other baselined work products including review procedures and change management procedures.

*Partial Example*

1. All work products will be stored in a centralized CVS repository running on a central server.
2. The naming convention for documents will be: NNN-VVV.suffix where NNN is a mnemonic that reflects the function of the document, VVV is a 3 digit version number, and 'suffix' is the standard/normal suffix for the document type. For example, the second version of the requirements document created as a Microsoft Word document might be labeled: REQ-002.doc.
3. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in the CVS repository but not all will be under change control (subject to formal change control procedures.) Only the system requirements, project plan and source code will be baselined and under configuration control.
4. Items that are subject to change control will be considered baselined after a group review at the end of the life cycle phase during which they are created. Baselined here means that the product has undergone a formal review and can only be changed through the prescribed change control procedures.
5. The change control procedure once a product is baselined is: (1) anyone wanting to make a change to a baselined item sends an email to the rest of the group describing the change, reason for the change, expected impact, and timeline for integrating the change. (2) if no one responds to the group within 2 days with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change. If anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice their opinion. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.
6. Including a change history with all documents is encouraged but only required for baselined documents. The change history should be at the front of the work item and include: (1) the name of the person making the change, (2) brief description of what has changed, (3) reason for the change, and (4) the date the change was integrated.

## Verification and Validation Plan

The verification and validation plan defines what actions are being taken to assure the quality of the development process and resulting software products.

*Partial Example*

The Verification and Validation plan is specified as a separate documented located in the version control system at: http://company.com/svn/project-name/docs/VandVPlan.doc

## Product Acceptance Plan

The product acceptance plan defines what is acceptable in terms of product quality and product functionality. Acceptance criteria should be objective and measurable. Note product success is one aspect of project success. Teams wanting to establish a clear understanding of what will be considered acceptable project performance may want to define a more general plan for project success that includes quantitative goals for delivery date, cost, etc.

*Partial Example*

The Verification and Valida