**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 5](#_Toc413399129)

[1.1 Purpose and Scope 5](#_Toc413399130)

[1.2 Goals and Objectives 5](#_Toc413399131)

[1.3 Project Deliverables 5](#_Toc413399132)

[1.4 Assumptions and Constraints 6](#_Toc413399133)

[1.5 Schedule and Budget Summary 6](#_Toc413399134)

[1.6 Success Criteria 6](#_Toc413399135)

[1.7 Definitions 6](#_Toc413399136)

[1.8 Evolution of the Project Plan 7](#_Toc413399137)

[2 Startup Plan 7](#_Toc413399138)

[2.1 Team Organization 7](#_Toc413399139)

[2.2 Project Communications 8](#_Toc413399140)

[2.3 Technical Process 8](#_Toc413399141)

[2.4 Tools 8](#_Toc413399142)

[3 Work Plan 9](#_Toc413399143)

[3.1 Release Plan 9](#_Toc413399144)

[3.2 Iteration Plans 9](#_Toc413399145)

[4 Control Plan 9](#_Toc413399146)

[4.1 Monitoring and Control 9](#_Toc413399147)

[4.2 Project Measurements 9](#_Toc413399148)

[5 Supporting Process Plans 10](#_Toc413399149)

[5.1 Risk Management Plan 10](#_Toc413399150)

[5.2 Configuration Management Plan 11](#_Toc413399151)

[5.3 Verification and Validation Plan 11](#_Toc413399152)

[5.4 Product Acceptance Plan 11](#_Toc413399153)

[5.5 Bibliography 11](#_Toc413399154)

# 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 on-call rotation schedules. Presently, management spends time and effort manually creating and maintaining these schedules. The On-Call Assistant will handle this tedious task, increasing workflow and decreasing 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

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 focus on project updates and agenda items created by the team leader. After weekly meetings team members should understand their tasks for the next iteration as well as the time-line for each 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, project documentation, organization and presentation of project information to the appropriate parties, and helping team members 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 account has been created for managing tasks, updating team members with changes, and 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. Wikipedia says about Scrum1:

*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.
* Source Control – 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

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.

|  |  |
| --- | --- |
| **Risk** | **Risk Response** |
| 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. |
| Unclear or vague requirements | 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. |

## Configuration Management Plan

1. All deliverables are stored on a distributed Git repository which can be cloned and modified by team members.
2. Only base lined documents will be subject to change control procedures.
3. Deliverables will be considered under change control after review by manager and team during a formal meeting. Once the state of the document is approved, it will be placed under change control.
4. For a change to occur, a team member must submit a change request to the GitHub repository. The request must include a change description, motivation, impact estimate, and timeline to implement.
5. Change requests will be discussed and reviewed by the team. The team manager has the final authority to approve/deny change requests.
6. To track and manage changes, appropriate tables will be updated in the project documentation for each change. We will track the team member who made the change, the date, and create an issue number corresponding to the GitHub issue to allow for traceability.

## Verification and Validation Plan

* Before committing a change to the team’s shared repository, the team member must test the functionality of the altered portion of the application. For example, if the scheduling algorithm is modified, a test schedule must be generated.
* All changes implemented during an iteration must be performed on a distinct development branch. The master branch is always a stable release.
* At the completion of the iteration, team members must functionally test their respective portion of the application. The manager will thoroughly test each of the implemented use cases.
* After testing and validation, changes will be merged into master for release.
* Unit testing is to be implemented on completed modules to ensure regression testing should future changes occur.

## Product Acceptance Plan

TBD

## Bibliography

1. "Scrum (software Development)." *Wikipedia*. Wikimedia Foundation, n.d. Web. 06 Mar. 2015. <http://en.wikipedia.org/wiki/Scrum\_%28software\_development%29>.