|  |
| --- |
| *VENI App* |
| **Software Project Management Plan** |
| **SE 6387 Advanced Software Engineering Project**  **R.Z. Wenkstern**    ***April 22, 2015*** |

|  |
| --- |
| **Group *X*** |
| **Brian MacKay** |
| **Anant Kambli** |
| **Raleigh Murráy** |
| **Shahed Shuman** |
| **Kathryn Whitmire** |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Authors** |
| 1.0 | 05-Feb-2015 | Initial draft | Group |
| 2.0 | 19-Feb-2015 | Revisions to scheduled dates, grammar fixes, and included MS Project Schedule | K. Whitmire,  A. Kambli |
| 3.0 | 03-Mar-2015 | Revise sections 1.4, 2.1, 3, 3.1, 3.3, 4.2, 4.4, 5.2, 5.4, 5.5, and 5.5.1 | K. Whitmire |
| 4.0 | 14-Apr-2015 | Minor grammatical changes, updates to schedule dates, revisions to section 5, date correction | K. Whitmire |

**Contents**

[Revision History 2](#_Toc416817082)

[1. Overview 5](#_Toc416817083)

[1.1 Purpose, Scope, and Objectives 5](#_Toc416817084)

[1.2 Assumptions and Constraints 5](#_Toc416817085)

[1.2.1 Assumptions 5](#_Toc416817086)

[1.3 Project Deliverables 6](#_Toc416817087)

[1.4 Schedule and Budget Summary 6](#_Toc416817088)

[2. Project Organization 7](#_Toc416817089)

[2.1 Roles and Responsibilities 7](#_Toc416817090)

[3. Managerial Process Plan 8](#_Toc416817091)

[3.1 Start-up Plan 8](#_Toc416817092)

[3.1.1 Estimation Plan 8](#_Toc416817093)

[3.1.2 Staffing Plan 8](#_Toc416817094)

[3.1.3 Resource Acquisition Plan 8](#_Toc416817095)

[3.2 Work Plan 9](#_Toc416817096)

[3.2.1 Work Activities 9](#_Toc416817097)

[3.2.2 Schedule/Resource/Budget Allocation 10](#_Toc416817098)

[3.3 Risk Management Plan 11](#_Toc416817099)

[3.3.1 Risk Identification 11](#_Toc416817100)

[3.3.2 Risk Analysis 12](#_Toc416817101)

[3.3.3 Risk Mitigation 12](#_Toc416817102)

[4. Technical Process Plan 13](#_Toc416817103)

[4.1 Process Model 13](#_Toc416817104)

[4.2 Methods, Tools, and Techniques 13](#_Toc416817105)

[4.3 Infrastructure Plan 13](#_Toc416817106)

[4.4 Product Acceptance Plan 14](#_Toc416817107)

[5. Supporting Process Plans 14](#_Toc416817108)

[5.1 Configuration Management Plan 14](#_Toc416817109)

[5.2 Test Plan 15](#_Toc416817110)

[5.2.1 Testing Scope 15](#_Toc416817111)

[5.2.2 Testing Strategy 16](#_Toc416817112)

[5.3 Documentation Plan 16](#_Toc416817113)

[5.4 Quality Assurance Plan 16](#_Toc416817114)

[5.5 Communications Management Plan 16](#_Toc416817115)

[5.5.1 Project Team Meetings 17](#_Toc416817116)

[Appendix A: Glossary 18](#_Toc416817117)

[Appendix B: References 19](#_Toc416817118)

# 1. Overview

The *Veni* Software Project Plan describes the *Veni* system and outlines the anticipated activities required to complete this effort by the scheduled end date.

## 1.1 Purpose, Scope, and Objectives

The purpose of *Veni* is intended to simplify the check-in process for veterans at Veteran Administration (VA) facilities. In order to accomplish this, the development team will analyze the requirements for, design, implement, and maintain the *Veni* system software.

All activities directly related to the purpose are considered to be in scope. This includes checking in for an appointment, getting directions to the VA facility, and providing appointment reminders.

The objectives of the project are as follows:

* Complete the project by the project due date (April 22nd, 2015)
* Complete the project within budget ($100)
* Provide all deliverables identified in section 1.1.3 by the project due date
* Fulfill all stated requirements (as found in the System Requirements Specification)

## 1.2 Assumptions and Constraints

### 1.2.1 Assumptions

The project will be planned with the following assumptions:

* This project will deliver the *Veni* Smartphone application software
* *Veni* will be portable to multiple Smartphone OS, but the prototype will target the Android OS.
* The *Veni* System Server will run in the J2EE server, running on Linux within a cloud-hosted virtual machine.
* VA firewall will permit application access to VA information
* During development, the *Veni* System Server will interact with a cloud-hosted *Veni*-specific VistA instance.
* The VA should allow us to interact with a VA-hosted “sandbox” VistA system.

**Constraints**

The project will have the following constraints:

* Reduce appointment wait time by 50%
* Complete this project by April 22nd 2015
* Accomplish this project with a 5 person team
* Gain the cooperation of the VA for data access (for future development)

## 1.3 Project Deliverables

The following items are the deliverables provided prior to the completion of the project.

* Software programs (source code and objects)
* *Veni* System Environment
* *Veni* System User Documentation
* Project documentation
  + Software Project Management Plan (SPMP)
  + Software Requirements Specification (SRS)
  + Software Design Document (SDD)
  + Software Testing Documentation (STD)
* Commitments
  + Fortnightly Status Reports

## 1.4 Schedule and Budget Summary

The budget for this project is flexible, but it is desired that all costs will be comprised of time and effort. The monetary cost will be $100 or less due to the use of free and open source tools. The effort involved for each team member will be at least 20 hours of time per week for the 10 week duration of the project, with a maximum of 30 hours of time per week. Cumulatively, this will be 200 to 300 hours of each individual’s time over the 10 week project (1000 to 1500 hours total).

The project has the following high-level schedule:

* Delivery of baseline project plan: February 5, 2015
* Software products ready for operation: April 22, 2015

# 2. Project Organization

A team of five resources are assigned to this project. The entire project team is responsible for the successful delivery of the project. The following five people form the project team.

* Anant Kambli
* Brian MacKay
* Raleigh Murráy
* Shahed Shuman
* Kathryn Whitmire

The team members are individually responsible for handling coordination between each other in order to complete their given tasks, integrating the separate tasks, and submitting the deliverables. Team members will work on their tasks as per discussion in the team meetings.

## 2.1 Roles and Responsibilities

**Analyst/Designers** – Analyst/Designers are responsible for gathering requirements from a user perspective and architecture design.

**Programmers** – Programmers are responsible for coding and unit testing of applications.

**Testers** – Testers are responsible for performing integration testing as well as conducting user acceptance tests (UAT).

**Technical Writer** – Technical writer is responsible for writing the user manual as well as testing the usability of the application.

**Project Manager** - Project Manager is responsible for managing the overall project and updating the project status via status reports.

All of the aforementioned roles will be fulfilled by each of the team members at different times during the project.

# 3. Managerial Process Plan

This section details the process for how the manager will handle schedule deviations during the project lifecycle in order to deliver a software product that will satisfy the needs of the client as outlined in the SRS.

## 3.1 Start-up Plan

The following section addresses the resources and materials needed to start the project and will include: the estimation plan, the staffing plan, and the resource acquisition plan.

### 3.1.1 Estimation Plan

The team is using a time-boxed agile process, and therefore have established initial estimates and plan to update them as the project progresses based on each individual’s assessment of their workload.

### 3.1.2 Staffing Plan

Due to limited resources, all team members will take on all roles at various stages of the project (as mentioned in section 2.1). This is in keeping with the agile philosophy that the project follows. The following table illustrates how the team’s time will be spent in each role:

|  |  |
| --- | --- |
| Resource Name | Max. Units\* |
| Analysts/Designers | 20% |
| Programmers | 25% |
| Project Manager | 10% |
| Technical Writer | 25% |
| Testers | 20% |

\*Units refers to the percentage of time utilized from the team (e.g. 20% means 20% of the total team effort is spent on a task; for example, 20% from each person or 25% from four members while the fifth works on another task)

### 3.1.3 Resource Acquisition Plan

The project team shall be responsible for acquiring all non-human resources required by the project. The non-human resources required for the project are:

* *Veni* System Server
* Linux Virtual Machine
* GitHub Software repository

## 3.2 Work Plan

The SPMP will specify the work activities, schedule, and resources for this project.

### 3.2.1 Work Activities

The overall project plan for the *Veni* team is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | Duration | Start | Finish | % Complete |
| **Veni System Project** | **65 days** | **Thu 1/22/15** | **Wed 4/22/15** | **100%** |
| **Initial Documentation** | **21 days** | **Thu 1/22/15** | **Thu 2/19/15** | **100%** |
| Executive Summary (Final) | 11 days | Thu 1/22/15 | Thu 2/5/15 | 100% |
| Feasibility Document (Final) | 11 days | Thu 1/22/15 | Thu 2/5/15 | 100% |
| Vision Document (Final) | 11 days | Thu 1/22/15 | Thu 2/5/15 | 100% |
| System Requirements Specification  (Initial Draft) | 21 days | Thu 1/22/15 | Thu 2/19/15 | 100% |
| SW Project Plan (Initial Draft) | 21 days | Thu 1/22/15 | Thu 2/19/15 | 100% |
| **High Level Design** | **11 days** | **Thu 2/5/15** | **Thu 2/19/15** | **100%** |
| Architecture Documentation | 11 days | Thu 2/5/15 | Thu 2/19/15 | 100% |
| Use Case Level 0 diagram | 11 days | Thu 2/5/15 | Thu 2/19/15 | 100% |
| Use Cases Fully Dressed | 11 days | Thu 2/5/15 | Thu 2/19/15 | 100% |
| **Agile Iterations** | **51 days** | **Thu 2/5/15** | **Thu 4/16/15** | **100%** |
| **Iteration 0: Set-up** | **11 days** | **Thu 2/5/15** | **Thu 2/19/15** | **100%** |
| Set up environment | 11 days | **Thu 2/5/15** | **Thu 2/19/15** | 100% |
| **Iteration 1: Detailed Design** | **3 days** | **Fri 2/20/15** | **Tue 2/24/15** | **100%** |
| Grey Box Sequence Diagram | 3 days | **Fri 2/20/15** | **Tue 2/24/15** | 100% |
| Subsystem Architecture  with operations diagram | 3 days | **Fri 2/20/15** | **Tue 2/24/15** | 100% |
| **Iteration 2: Check-in at Facility** | **30 days** | **Wed 2/25/15** | **Tue 4/7/15** | **100%** |
| **Detailed Design** | **16 days** | **Wed 2/25/15** | **Wed 3/18/15** | **100%** |
| White Box Sequence Diagram | 5 days | **Wed 2/25/15** | **Tue 3/3/15** | 100% |
| Design Class Diagram | 16 days | **Wed 2/25/15** | **Wed 3/18/15** | 100% |
| Implementation | 14 days | **Thu 3/19/15** | **Tue 4/7/15** | 100% |
| Testing (Unit) | 6 days | **Tue 3/31/15** | **Tue 4/7/15** | 100% |
| **Iteration 2: First Run Experience** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 3: Download Appointments** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 4: Verify Appointment Status** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 5: Transfer to Phone Calendar** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 6: Authenticate User** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 7: Get Directions to Facility** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Iteration 8: Maintain Administrative Information** | **7 days** | **Wed 4/8/15** | **Thu 4/16/15** | **100%** |
| Implementation | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| Testing (Unit) | 7 days | **Wed 4/8/15** | **Thu 4/16/15** | 100% |
| **Living Documentation** | **40 days** | **Fri 2/20/15** | **Thu 4/16/15** | **100%** |
| SW Project Plan (Intermediate) | 40 days | **Fri 2/20/15** | **Thu 4/16/15** | 100% |
| SRS Document (Intermediate) | 40 days | **Fri 2/20/15** | **Thu 4/16/15** | 100% |
| Design Documents (Intermediate) | 40 days | **Fri 2/20/15** | **Thu 4/16/15** | 100% |
| **Integration** | **4 days** | **Fri 4/17/15** | **Wed 4/22/15** | **100%** |
| Integration Testing | **4 days** | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| Auto Testing | **4 days** | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| Test Documentation | **4 days** | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| **Validation** | **24 days** | **Fri 3/20/15** | **Wed 4/22/15** | **100%** |
| System Testing | 20 days | **Fri 3/20/15** | **Thu 4/16/15** | 100% |
| **Finalize Documentation** | **4 days** | **Fri 4/17/15** | **Wed 4/22/15** | **100%** |
| SW Project Plan (Final) | 4 days | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| SRS Document (Final) | 4 days | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| Design Documents (Final) | 4 days | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| Testing Document (Final) | 4 days | **Fri 4/17/15** | **Wed 4/22/15** | 100% |
| User Manual Document (Final) | 4 days | **Fri 4/17/15** | **Wed 4/22/15** | 100% |

### 3.2.2 Schedule/Resource/Budget Allocation

The schedule for each team member will be established for each agile iteration of the project. Additionally the team will make adjustments to the schedule in order to balance the workload amongst the team (See previous section 3.2.1).

The resources for this project shall be allocated on an as needed basis. Resources are as follow:

|  |  |
| --- | --- |
| Resource Name | Max. Units |
| Brian MacKay | 100% |
| Anant Kambli | 100% |
| Raleigh Murráy | 100% |
| Shahed Shuman | 100% |
| Kathryn Whitmire | 100% |

\*Units refers to the percentage of effort for each team member (e.g. 50% would mean the individual could only spend half their available time on this effort)

The budget for this project shall remain under $100.

## 3.3 Risk Management Plan

This section defines how risks associated with this project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project, and it provides templates and practices for recording and prioritizing risks by the Risk Manager and/or Risk Management Team.

### 3.3.1 Risk Identification

The project management team will identify risks by taking into account the project scope, schedule, cost, and quality. The following risks are identified by the project team:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Description** | **Solution** |
| GitHub unavailability | Low | Medium | Unavailability of source control system (GitHub) | All important documents and messages are sent via email. |
| Team member absence | Medium | Medium | Team member is unable to come to team meetings. | Email summary of accomplishments to the missing team member. |
| Unfamiliar with technique | High | High | Team member is not familiar with tools or techniques used in this project. | Do team tutorial or self-learning. Evaluate team member skills before each phase and/or re-assign team roles. |
| Reuse | High | Medium | Team is not able to reuse code and/or documentation. | Only reuse if parts are well understood, otherwise create it from scratch. |
| Lack of communication | High | High | Team members fail to communicate at critical times | Be active and effective in communication, monitor team member response, and project involvement. Provide friendly environment. |

### 3.3.2 Risk Analysis

All risks will be assessed to identify the range of possible project outcomes. Risks will be prioritized by their level of importance, and shall be categorized as follows:

**Probability**

* High – Greater than 70% probability of occurrence
* Medium – Between 30% and 70% probability of occurrence
* Low – Below 30% probability of occurrence

**Impact**

* High – Risk that has the potential to greatly impact project cost, project schedule or performance
* Medium – Risk that has the potential to slightly impact project cost, project schedule or performance
* Low – Risk that has relatively little impact on cost, schedule or performance.

### 3.3.3 Risk Mitigation

The project management team comes with the following risk mitigation ideas for this project:

* A lack of programmer availability would have a great impact on the overall success of the project. To mitigate this risk, programmers will use a common code repository for code storage and version control so another team member can continue work.
* A lack of participation or unforeseen absence by a team member would also have a high impact on project success. To mitigate this risk, team members will put in extra hours and share tasks.  However, all members are highly motivated and actively engaged in the development of the project plan, so this has a low chance of occurrence.

# 4. Technical Process Plan

The following section will cover the plans used to develop the work products, project infrastructure, and *Veni* system acceptance plan.

## 4.1 Process Model

The *Veni* project team will follow an incremental and iterative development model for its deliverables.

The initial roles will be selected at the start of the project. Roles will then rotate among the team members depending on their assigned tasks, thus providing the opportunity for each team member to have more than one role during the course of the project.

## 4.2 Methods, Tools, and Techniques

The project will use object-oriented analysis and design coupled with modern agile development methods to deliver the *Veni* system software, with work activities organized in the Work Breakdown Structure (WBS) as shown in section 3.2.2.

The tools used during this project includes:

* MS Word 2013
* MS PowerPoint 2013
* MS Project 2013
* GitHub
* Visio
* Intel XDK

## 4.3 Infrastructure Plan

The *Veni* system will contain the *Veni* phone application and the *Veni* System Server. The system will be developed for portability in order for it to run on multiple smartphone operating systems. However, during the prototype development, this will likely be restricted to a single platform.

The *Veni* phone app will connect to the *Veni* System Server. The server will run a J2EE instance within a Linux virtual machine and will be hosted in Amazon’s commercial cloud infrastructure. The phone application will communicate with the system server using a simple JSON + REST interface.

## 4.4 Product Acceptance Plan

Every project deliverable will be accepted formally by the team by signing off on the documentation. At the end of the project, the team will install the product and perform an acceptance test.

The acceptance test will require the following features:

* Log-in
* Registration
* Download appointments
* Check-In
* Verify appointments
* Get directions to facility

In addition to the *Veni* system, the customer will expect the following documents:

* Executive Summary
* Feasibility Document
* Vision Document
* Software Requirements Specification
* Software Project Plan
* Architecture Design Document
* Testing Document
* *Veni* User Documentation
* *Veni* source code

# 5. Supporting Process Plans

## 5.1 Configuration Management Plan

To manage and maintain the source code, the project team will be using a source code control tool called GitHub. This allows different developers to work on the same items, with reduced chance of overwriting another’s work. This tool also tracks which changes were made, who made them, when they were made, and why. Finally, GitHub includes the ability to group versioned files as a single release, maintain multiple active releases concurrently (*branching*), and join different releases (*merging*).

The lists of documents that will be created and maintained under version control include:

* Software Project Management Plan (SPMP)
* Software Requirements Specification (SRS)
* Software Design Document (SDD)
* Software Testing Document (STD)
* Fortnightly Status Report (FSR)
* *Veni* User Documentation
* *Veni* source code

## 5.2 Test Plan

The team plans to create an end-to-end test following the happy-path, unit tests for each use case, and tests specifically for finding security issues. User acceptance testing will occur after the final release of the project.

### 5.2.1 Testing Scope

**List of features in testing scope:**

* First-Run experience
* Downloading appointments
* Updating the list of facilities
* Adding appointments to calendar (as in sending the list to phone calendar service)
* Getting directions to the clinic (as in sending the facility’s latitude/longitude coordinates to phone map service)
* Check-in
* End-to-end happy-path of system (failure to be tested in later releases)

**List of features out of testing scope:**

* When adding appointments to calendar, it is assumed the phone service is correct
* When getting directions to the clinic, it is assumed the phone service is correct
* Non-happy-path scenarios (this is to be tested in future phases)

### 5.2.2 Testing Strategy

The following testing will be done:

* Unit Testing (UT)
* System Integration Testing (SIT)
* User Acceptance Testing (UAT)
* Security Testing (ST)

## 5.3 Documentation Plan

There are a number of documents that will be produced during the lifetime of the project. All documents are the responsibility of the team members, who will discuss and review each document before their baseline versions are issued and shared on the GitHub repository.

The following tools will be used:

* Visio (for UML Documentation)
* MS Word (for Project Documents)
* GitHub (for Document Repository)

The lists of documents that will be created and maintained under version control include:

* Software Project Management Plan (SPMP)
* Software Requirements Specification (SRS)
* Software Design Document (SDD)
* Software Testing Document (STD)
* Fortnightly Status Report (FSR)
* *Veni* User Documentation

## 5.4 Quality Assurance Plan

The project team will check that software products are reviewed, verify the results, and report issues. These reported problems will be resolved by the team in accordance with the technical requirements.

## 5.5 Communications Management Plan

The following section addresses how the team members communicate with each other.

### 5.5.1 Project Team Meetings

The team will conduct meetings at UTD Campus as needed, but are currently planned on a fortnightly basis. Besides meeting at UTD, the team will text and use e-mail to contact each other to share any information related to this project. Conference Bridge will be used to have teleconference meetings if required.

# Appendix A: Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| VA | Veterans Administration |
| *Veni* | Name of the system application |
| VISTA | Veterans Health Information Systems and Technology Architecture |
| PHI | Protected Health Information |
| PII | Personally identifiable information |
| HIPAA | Health Insurance Portability and Accountability Act |
| IEEE | Institute of Electrical and Electronics Engineers |
| SPMP | Software Project Management Plan |
| SRS | Software Requirements Specification |
| BWSR | Bi-Weekly Status Report |
| IEEE 1058-  1998 | the IEEE standard for Software Project Management Plans on which  this plan is based |
| VM | Virtual Machine |

# Appendix B: References

**1.) www.va.gov**