|  |
| --- |
| *VENI App* |
| **Software Project Management Plan** |
| **SE 6387 Advanced Software Engineering Project**  **R.Z. Wenkstern**    ***February 5, 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 | Kathryn Whitmire, Anant Kambli |

**Contents**

[Revision History 2](#_Toc410570225)

[1. Overview 5](#_Toc410570226)

[1.1 Purpose, Scope, and Objectives 5](#_Toc410570227)

[1.2 Assumptions and Constraints 5](#_Toc410570228)

[1.2.1 Assumptions 5](#_Toc410570229)

[1.3 Project Deliverables 6](#_Toc410570230)

[1.4 Schedule and Budget Summary 6](#_Toc410570231)

[2. Project Organization 7](#_Toc410570232)

[2.1 Roles and Responsibilities 7](#_Toc410570233)

[3. Managerial Process Plan 8](#_Toc410570234)

[3.1 Start-up Plan 8](#_Toc410570235)

[3.1.1 Estimation Plan 8](#_Toc410570236)

[3.1.2 Staffing Plan 8](#_Toc410570237)

[3.1.3 Resource Acquisition Plan 9](#_Toc410570238)

[3.2 Work Plan 9](#_Toc410570239)

[3.2.1 Work Activities 9](#_Toc410570240)

[3.2.2 Schedule/Resource/Budget Allocation 9](#_Toc410570241)

[3.3 Risk Management Plan 10](#_Toc410570242)

[3.3.1 Risk Identification 10](#_Toc410570243)

[3.3.2 Risk Analysis 11](#_Toc410570244)

[3.3.3 Risk Mitigation 11](#_Toc410570245)

[4. Technical Process Plan 12](#_Toc410570246)

[4.1 Process Model 13](#_Toc410570247)

[4.2 Methods, Tools, and Techniques 13](#_Toc410570248)

[4.3 Infrastructure Plan 13](#_Toc410570249)

[4.4 Product Acceptance Plan 14](#_Toc410570250)

[5. Supporting Process Plans 14](#_Toc410570251)

[5.1 Configuration Management Plan 14](#_Toc410570252)

[5.2 Test Plan 15](#_Toc410570253)

[5.2.1 Testing Scope 15](#_Toc410570254)

[5.2.2 Testing Strategy 15](#_Toc410570255)

[5.3 Documentation Plan 16](#_Toc410570256)

[5.4 Quality Assurance Plan 16](#_Toc410570257)

[5.5 Communications Management Plan 16](#_Toc410570258)

[5.5.1 Project Team Meetings 17](#_Toc410570259)

[Appendix A: Glossary 18](#_Toc410570260)

[Appendix B: References 19](#_Toc410570261)

# 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 20th, 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 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 20th 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 Manual
* 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 project has the following high-level schedule:

* Delivery of baseline project plan: February 5, 2015
* Software products ready for operation: April 20, 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 usability of the application.

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

# 3. Managerial Process Plan

The SPMP will specify the project management processes for the project and will include: the project startup plan, project work plan, and risk management plan.

In order to be successful, the team must deliver a software product that will satisfy the needs of the client as outlined in the SRS.

## 3.1 Start-up Plan

The SPMP outlines 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. This is in keeping with the agile philosophy that the project follows.

|  |  |
| --- | --- |
| Resource Name | Max. Units |
| Analysts/Designers | 100% |
| Programmers | 125% |
| Project Manager | 50% |
| Technical Writer | 125% |
| Testers | 100% |

### 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 |
| **Veni System Project** | **63 days?** | **Thu 1/22/15** | **Mon 4/20/15** |
| **Initial Documentation** | **21 days?** | **Thu 1/22/15** | **Thu 2/19/15** |
| Executive Summary | 11 days? | Thu 1/22/15 | Thu 2/5/15 |
| Feasibility Document | 11 days? | Thu 1/22/15 | Thu 2/5/15 |
| Vision Document | 11 days? | Thu 1/22/15 | Thu 2/5/15 |
| System Requirements Specification (ver 1) | 21 days? | Thu 1/22/15 | Thu 2/19/15 |
| SW Project Plan (ver 1) | 21 days? | Thu 1/22/15 | Thu 2/19/15 |
| **High Level Design** | **11 days?** | **Thu 2/5/15** | **Thu 2/19/15** |
| Architecture Documentation | 11 days? | Thu 2/5/15 | Thu 2/19/15 |
| Use Case Level 0 diagram | 11 days? | Thu 2/5/15 | Thu 2/19/15 |
| Use Cases Fully Dressed | 11 days? | Thu 2/5/15 | Thu 2/19/15 |
| **Agile Iterations** | **46 days?** | **Thu 2/5/15** | **Thu 4/9/15** |
| **Iteration 0: Set-up** | **11 days?** | **Thu 2/5/15** | **Thu 2/19/15** |
| Set up environment | 11 days? | **Thu 2/5/15** | **Thu 2/19/15** |
| **Iteration 1: Check-in at Facility** | **5 days?** | **Fri 2/20/15** | **Thu 2/26/15** |
| **Detailed Design** | **5 days?** | **Fri 2/20/15** | **Thu 2/26/15** |
| Sequence Diagram | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Design Class Diagram | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Implementation | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Testing (Unit) | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| **Iteration 2: Maintain Administrative Information** | **5 days?** | **Fri 2/20/15** | **Thu 2/26/15** |
| **Detailed Design** | **5 days?** | **Fri 2/20/15** | **Thu 2/26/15** |
| Sequence Diagram | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Design Class Diagram | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Implementation | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| Testing (Unit) | 5 days? | **Fri 2/20/15** | **Thu 2/26/15** |
| **Iteration 3: First Run Experience** | **5 days?** | **Fri 2/27/15** | **Thu 3/5/15** |
| **Detailed Design** | **5 days?** | **Fri 2/27/15** | **Thu 3/5/15** |
| Sequence Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Design Class Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Implementation | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Testing (Unit) | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| **Iteration 4: Download Appointments** | **5 days?** | **Fri 2/27/15** | **Thu 3/5/15** |
| **Detailed Design** | **5 days?** | **Fri 2/27/15** | **Thu 3/5/15** |
| Sequence Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Design Class Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Implementation | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Testing (Unit) | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| **Iteration 5: Get Directions to Facility** | **5 days?** | **Fri 3/6/15** | **Thu 3/12/15** |
| **Detailed Design** | **5 days?** | **Fri 3/6/15** | **Thu 3/12/15** |
| Sequence Diagram | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Design Class Diagram | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Implementation | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Testing (Unit) | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| **Iteration 6: Verify Appointment Status** | **5 days?** | **Fri 3/6/15** | **Thu 3/12/15** |
| **Detailed Design** | **5 days?** | **Fri 3/6/15** | **Thu 3/12/15** |
| Sequence Diagram | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Design Class Diagram | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Implementation | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| Testing (Unit) | 5 days? | **Fri 3/6/15** | **Thu 3/12/15** |
| **Iteration 7: Transfer to Phone Calendar** | **15 days?** | **Fri 2/27/15** | **Thu 3/19/15** |
| **Detailed Design** | **5 days?** | **Fri 2/27/15** | **Thu 3/5/15** |
| Sequence Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Design Class Diagram | 5 days? | **Fri 2/27/15** | **Thu 3/5/15** |
| Implementation | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| Testing (Unit) | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| **Iteration 8: Transfer to Phone Calendar** | **5 days?** | **Fri 3/13/15** | **Thu 3/19/15** |
| **Detailed Design** | **5 days?** | **Fri 3/13/15** | **Thu 3/19/15** |
| Sequence Diagram | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| Design Class Diagram | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| Implementation | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| Testing (Unit) | 5 days? | **Fri 3/13/15** | **Thu 3/19/15** |
| **Iteration 9: Authenticate User** | **5 days?** | **Fri 3/20/15** | **Thu 3/26/15** |
| **Detailed Design** | **5 days?** | **Fri 3/20/15** | **Thu 3/26/15** |
| Sequence Diagram | 5 days? | **Fri 3/20/15** | **Thu 3/26/15** |
| Design Class Diagram | 5 days? | **Fri 3/20/15** | **Thu 3/26/15** |
| Implementation | 5 days? | **Fri 3/20/15** | **Thu 3/26/15** |
| Testing (Unit) | 5 days? | **Fri 3/20/15** | **Thu 3/26/15** |
| **Living Documentation** | **35 days?** | **Fri 2/20/15** | **Thu 4/9/15** |
| SW Project Plan (Intermediate) | 35 days? | **Fri 2/20/15** | **Thu 4/9/15** |
| SRS Document (Intermediate) | 35 days? | **Fri 2/20/15** | **Thu 4/9/15** |
| **Integration** | **12 days** | **Fri 3/27/15** | **Mon 4/13/15** |
| Integration Testing | 12 days | **Fri 3/27/15** | **Mon 4/13/15** |
| Auto Testing | 12 days | **Fri 3/27/15** | **Mon 4/13/15** |
| Test Documentation | 12 days | **Fri 3/27/15** | **Mon 4/13/15** |
| **Validation** | **23 days?** | **Thu 3/19/15** | **Mon 4/20/15** |
| System Testing | 18 days? | Thu 3/19/15 | **Mon 4/13/15** |
| **Finalize Documentation** | **5 days?** | **Tue 4/14/15** | **Mon 4/20/15** |
| SW Project Plan (Final) | 5 days? | **Tue 4/14/15** | **Mon 4/20/15** |
| SRS Document (Final) | 5 days? | **Tue 4/14/15** | **Mon 4/20/15** |

### 3.2.2 Schedule/Resource/Budget Allocation

The schedule for each team member will be established 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% |

The budget for this project shall remain under $100.

## 3.3 Risk Management Plan

This Risk Management Plan 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 SPMP 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 the agile software development methodology 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 deliverable of the project will be accepted formally by the team member by signing off on the documentation. At the end of the project, the team will install the product and perform an acceptance test. This may result in additional requests for change and improvements.

# 5. Supporting Process Plans

## 5.1 Configuration Management Plan

The purpose of the configuration management plan is to provide the basic guidelines and procedures for administering source code, producing software development builds, controlling change, and managing software configurations. Specifically, this plan ensures the integrity, reliability, and reproducibility of developing software products from conception to release.

A Configuration Item (CI) is an entity designated for configuration management, which consists of multiple related work products that form a baseline. This logical grouping provides ease of identification and controlled access. These work products include source code, executables, and project deliverables.

The configuration control process evaluates, approves, and manages changes to controlled items. This includes tracking the configuration of each of the CIs, approving a new configuration, and updating the baseline.

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*).

## 5.2 Test Plan

The purpose of the test plan is to describe the scope, approach, resources, and schedule of intended test activities. It covers test cases, the testing approach, and the acceptance criteria. It is a record of the test planning process.

### 5.2.1 Testing Scope

Identify all *Veni* System features that will be tested.

**List of features in testing scope:**

**TBD at later revision of document (Current projection: 2/26/15)**

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

**TBD at later revision of document (Current projection: 4/13/15)**

### 5.2.2 Testing Strategy

The testing strategy describes the overall testing approach. For each major group of *Veni* system features, it will specify the major activities, techniques, and tools used to ensure adequate testing of the designated groups of features. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one.

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 Manual

## 5.4 Quality Assurance Plan

The goal of the Quality Assurance (QA) plan is to document procedures and activities related to quality assurance. It verifies that all software and documentation meet all technical requirements. The QA procedures will examine all deliverables to determine compliance with technical and performance requirements.

The project team will check that software products are reviewed, verify 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 Communications Management Plan (CMP) 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. 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**