Software Requirements Specification

for

SMART Agile

Version 0.1

Prepared By :

Monika Khanna

Manas Pradhan

Suzy

Anusha

Ashesh

Prepared for

SEN 941-6, Software Engineering

Instructor: Dr.Srinivasan P Mandyam, Ph.D.

Spring 2015

05-07-2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Description** | **Version** |
| 04-10-2015 | Monika Khanna | Initial Version | 0.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History 2

Document Approval 2

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 References 5

1.5 Overview 5

2. General Description 5

2.1 Product Perspective 5

2.2 Product Functions 6

Project Management 6

2.3 User Characteristics 6

2.4 General Constraints 6

2.5 Assumptions and Dependencies 6

3. Specific Requirements 6

3.1 External Interface Requirements 6

3.1.1 User Interfaces User Interfaces 6

3.1.2 Hardware Interfaces 6

3.1.3 Software Interfaces 6

3.1.4 Communications Interfaces 6

3.2 Functional Requirements 6

3.2.1 Project Management 6

3.2.2 Product Planning 7

3.2.2 Sprint Planning 7

3.2.2 Sprint Tracking 7

3.2.2 Generate Reports 8

3.3 Use Cases 8

3.3.1 Use Case #1 8

3.3.2 Use Case #2 8

3.4 Classes / Objects 9

3.4.1 <Class / Object #1> 9

3.4.2 <Class / Object #2> 9

3.5 Non-Functional Requirements 9

3.5.1 Performance 9

3.5.2 Reliability 9

3.5.3 Availability 9

3.5.4 Security 9

3.5.5 Maintainability 9

3.5.6 Portability 9

3.6 Inverse Requirements 9

3.7 Design Constraints 9

3.8 Logical Database Requirements 9

3.9 Other Requirements 9

4. Change Management Process 10

A. Appendices 10

A.1 Appendix 1 10

A.2 Appendix 2 10

# 1. Introduction

## 1.1 Purpose

This document provides software requirements for developing a web based Agile project management tool. Hence forth the tool will be referred to as SMART Agile.

## 1.2 Scope

SMART Agile product shall provide features that will enable software development teams to manage projects that follow Agile Scrum Framework. This tool is intended to be used by Scrum Teams- Product Owners, Scrum Master , developers, testers and other stakeholders that are part of Scrum team.

SMART Agile shall serve following purpose for various intended users of the product:

1. Product Managers:

Product Managers shall use this product to manage requirements(user stories) for customer specific projects or a new product initiative and to plan high level product/project roadmap. It shall serve as Requirement Management Tool for the product managers.

1. Scrum Master

Scrum Master shall use this product to estimate stories, create/manage project specific stories that are needed to meet the requirements of user stories , to plan/manage project iterations (called sprints in Scrum framework) consisting of a set of stories.

1. Developers/testers and other stakeholders

Developers/testers/other stakeholders shall use this tool to report progress on various stories allocated to them as part of sprint.

The tool shall also generate various kinds of reports that assist the Scrum teams to provide sprint completion forecast and progress status.

## 1.3 Definitions, Acronyms, and Abbreviations

*Agile*

*Product Owner*

*Project Specific Story*

*Scrum*

*Scrum Master*

*Sprint*

*User Story*

*Work Item*

## 1.4 References

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*

*(2) Identify each document by title, report number - if applicable - date, and publishing organization.*

*(3) Specify the sources from which the references can be obtained.*

*This information may be provided by reference to an appendix or to another document.*

## 1.5 Overview

*This subsection should:*

*(1) Describe what the rest of the SRS contains*

*(2) Explain how the SRS is organized.*

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

## 2.2 Product Functions

### Project Management

## 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

## 2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces User Interfaces

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

### 3.2.1 Setup Project

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| P | Create Project |
|  | Modify Project Settings |
|  | Create Team |
|  | Edit Team |
|  | Delete Team |
|  | Create Team Members |
|  | Edit Team Member |
|  | Delete Team Member |

### 3.2.2 Product Planning

**Backlog Management**

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| P | Create Backlog item |
|  | Prioritize Back log Items |
|  | Edit Backlog Items |
|  | Delete Backlog Items |
|  | Filter Backlog Items |
|  | Sort Backlog Items |

1. **Epic Management**

|  |  |
| --- | --- |
|  | Create epic |
|  | Delete Epic |
|  | Edit Epic |
|  |  |

### 3.2.2 Sprint Planning

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| P | Create Sprint |
|  | Edit Sprint |
|  | Delete Sprint |
|  | Add Backlog Items to Sprint |
|  | Remove Backlog Items from Sprint |
|  | Add Tasks to Sprint Backlog Items |
|  | Delete Tasks From Sprint Backlog Items |
|  | Edit Tasks |

### 3.2.2 Sprint Tracking

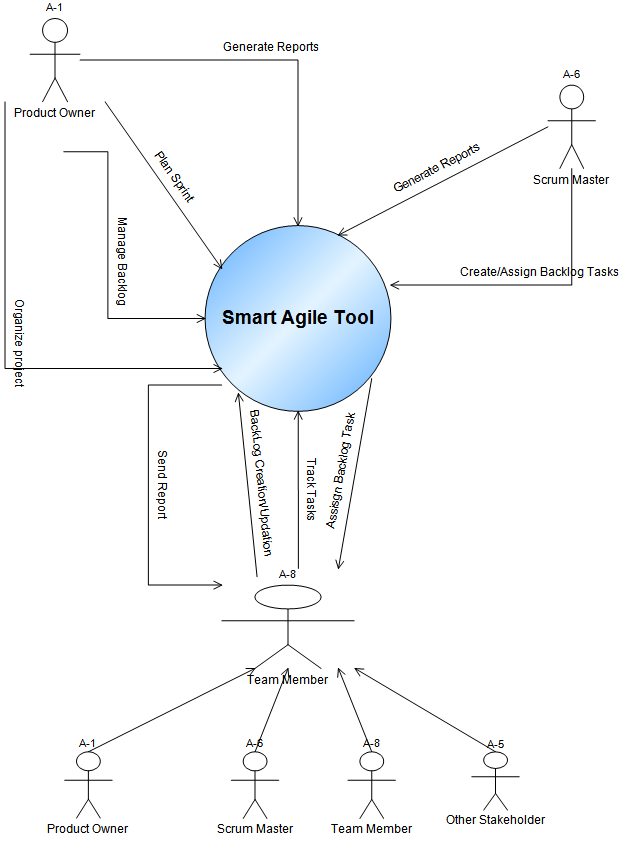
|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| P | Story Board Display  Completed  -Display all Planned Sprints in hierarchical manner  Top level Sprint  Stories In sprint  Tasks In Sprint  -Display Tasks in each Story in form of Swim Lane with 3 vertical columns  To do  In Progress  Completed  -Display Story Status with different colors  -Display Sprint Status with different colors |
|  | Track Tasks Progress. As tasks are tracked they can be moved to one of three states  -To Do  -In Progress  -Completed |
|  | Track Story Progress. States:  -To Do  -In progress  -Completed |
|  | Track Sprint Progress. States:  -In Progress  -Completed |

### 3.2.2 Report Generation

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| P | Generate Velocity Chart |
|  | Generate Burndown Chats for each sprint |

## 3.3 Use Cases

### 3.3.1 Project Context Diagram



### 3.3.1 Setup Project

### 3.3.2 Product Management

**BackLog Management**

**Epic Management**

### 3.3.2 Plan Sprint

### 3.3.2 Track Sprint

### 3.3.2 Generate Reports

…

## 3.4 Classes / Objects

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.5.1 Performance

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Inverse Requirements

State any \*useful\* inverse requirements.

## 3.7 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.9 Other Requirements

Catchall section for any additional requirements.

# 4. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2