Software Requirements Specification - SCRUM Project Management Application

Andrew Johnson and Max Warren

October 3, 2011

Contents

Revision History

Name	Date	Reason for Changes	Version
Andrew Johnson	9/27/11	initial draft	1.0 draft 1
Andrew Johnson and Max Warren	10/3/11	changes after inspection with team	1.0 approved

1 Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the SCRUM Project Management System (SPMS). This document is intended to be used by the members of the project team that will implement and verify correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Project Scope and Product Features

The SPMS will allow teams to plan their projects, create sprints, user stories, and tasks for those projects, and visualize project status using the SCRUM development methodology. A detailed project description is available in the *Vision and Scope Document for SCRUM Project Management System*[1]. The section in that document titled "Scope of Initial and Subsequent Releases" lists the features that are scheduled for full or partial implementation in this release.

1.3 References

1. Johnson, Andrew and Max Warren. *Vision and Scope Document for SCRUM Project Management System*

2. Microsoft. *Windows User Experience Interaction Guidelines*. `http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx`.

3. Schwaber, Ken and Mike Beedle. *Agile Software Development with Scrum*. ISBN 0130676349.

2 Overall Description

2.1 Product Perspective

The SCRUM Project Management System is a new system that replaces the current manual, whiteboard and sticky note-based processes for using the SCRUM development methodology. The context diagram in 1 illustrates the external entities and system interfaces for release 1.0.

2.2 User Classes and Characteristics

- Employees (favored): An employee is a member of a team using the SCRUM development methodology. There are over 9000 employees (including development, QA, and documentation) in 50 teams, all of which are currently using SCRUM. All these employees are expected to be using the SPMS daily. Employees will log their progress for the day. Each team of employees will view this progress daily during their stand-up meeting.

- Managers: A manager is either a team lead or a higher-level manager who is responsible for the overall direction of the project teams. There are about 70 managers in the company. The

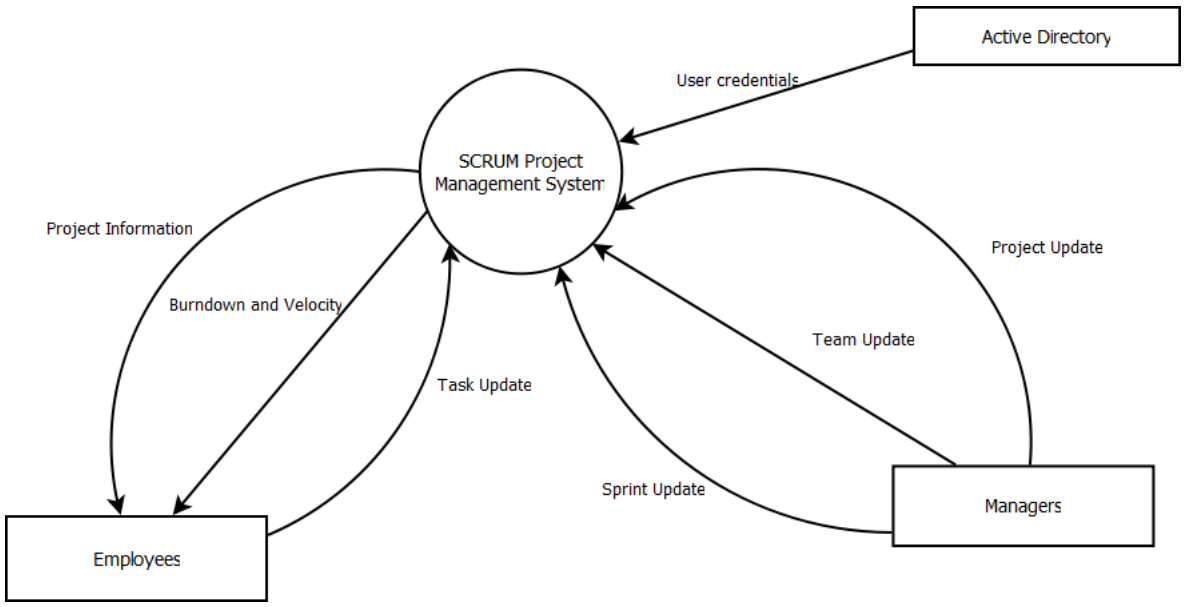

Figure 1: Context Diagram for Release 1.0 of the SCRUM Project Management System

50 team leads will review progress on their team's project in the SPMS system daily. The high-level managers will review the status of projects under their direction approximately once a week. Managers will create projects, sprints, and user stories in the system.

2.3 Operating Environment

OE-1: The SCRUM Project Management System client shall operate on the following operating systems: Windows XP, Windows Vista, Windows 7, and Windows 8.

OE-2: The database backend for the SCRUM Project Management System shall operate on a server running Windows Server 2008 with Microsoft SQL Server.

OE-3: The SCRUM Project Management System client shall permit user access to the database from the corporate Intranet and from an external Internet connection if the user is authorized to do so.

2.4 Design and Implementation Constraints

CO-1: The system shall use the current corporate standard Microsoft SQL Server DBMS.

CO-2: All C# code shall be written to run on the .NET 4.0 Framework.

2.5 User Documentation

UD-1: The system shall provide online hierarchical and cross-linked help document that describes and illustrates all system functions.

2.6 Assumptions and Dependencies

AS-1: Employees have access to Intranet-enabled computers on which to access the system.
DE-1: The system must be able to accept external authentication credentials for users.

3 System Features

3.1 Create a Project

3.1.1 Description and Priority

A Manager whose identity has been verified may create a new project in the system. Priority=High.

3.1.2 Stimulus/Response Sequences

Stimulus:	Manager requests to create a new project.
Response:	System queries Manager for details of project name and responsible team.
Stimulus:	Manager requests to change project details.
Response:	System queries Manager for updated project details.

3.1.3 Functional Requirements

- Project.Create: The system shall let an authenticated Manager create a new project.
- Project.Create.AssignTeam: The system shall prompt the Manager to assign a team to the project.
- Project.Create.SetName: The system shall prompt the Manager to give the project a name.
- Project.Create.SetStartDate: The system shall automatically set the project's start date to the current date.
- Project.Create.Save: The system shall let the Manager save the changes made to the new project.
- Project.Create.CreateBacklog: The system shall automatically create a backlog for user stories a project.

- Project.Create.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.
- Project.Update: The system shall let an authenticated Manager update an existing project.
- Project.Update.ChangeTeam: The system shall prompt the Manager to assign a new team to the project.
- Project.Update.ChangeName: The system shall prompt the Manager to give the project a new name.
- Project.Update.Save: The system shall let the Manager save the changes made to the project.
- Project.Update.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.

3.2 Create a Team

3.2.1 Description and Priority

A Manager whose identity has been verified may create a new team in the system. Priority=High.

3.2.2 Stimulus/Response Sequences

Stimulus:	Manager requests to create a new team.
Response:	System queries Manager for details of team name, team leader, and team members.
Stimulus:	Manager requests to change team details.
Response:	System queries Manager for updated team details.

3.2.3 Functional Requirements

- Team.Create: The system shall let an authenticated Manager create a new team.
- Team.Create.AssignLead: The system shall prompt the Manager to assign a leader to the team
- Team.Create.AssignMembers: The system shall prompt the Manager to assign Employees as members of the team.
- Team.Create.SetName: The system shall prompt the Manager to give the team a name.
- Team.Create.Save: The system shall let the Manager save the changes made to the new team.

- Team.Create.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.
- Team.Update: The system shall let an authenticated Manager update an existing team.
- Team.Update.ChangeTeamLead: The system shall prompt the Manager to assign a new lead to this team.
- Team.Update.ChangeMembers: The system shall prompt the Manager to change which Employees are members of this team
- Team.Update.ChangeName: The system shall prompt the Manager to give the team a new name.
- Team.Update.Save: The system shall let the Manager save the changes made to the team.
- Team.Update.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.

3.3 Create Sprint

3.3.1 Description and Priority

A Manager whose identity has been verified may create a new sprint for a project. Priority=High.

3.3.2 Stimulus/Response Sequences

Stimulus:	Manager requests to create a new sprint.
Response:	System queries Manager for the start date and end date.
Stimulus:	Manager requests to change sprint data.
Response:	System queries Manager for the updated start date or end date.

3.3.3 Functional Requirements

- Sprint.Create: The system shall let an authenticated Manager create a new sprint.
- Sprint.Create.SetEndDate: The system shall prompt the Manager for the end date of the sprint.
- Sprint.Create.SetStartDate: The system shall prompt the Manager for the start date of the sprint.
- Sprint.Create.Save: The system shall let the Manger save the changes made to the new sprint.

- Sprint.Create.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.
- Sprint.Update: The system shall let an authenticated Manager update an existing sprint.
- Sprint.Update.SetEndDate: The system shall prompt the Manager for the new end date of the sprint.
- Sprint.Update.SetStartDate: The system shall prompt the Manager for the new start date of the sprint.
- Sprint.Update.Save: The system shall let the Manger save the changes made to the sprint.
- Sprint.Update.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.

3.4 Add User Story

3.4.1 Description and Priority

A Manager whose identity has been verified may create a new user story for a project. Priority=High.

3.4.2 Stimulus/Response Sequences

Stimulus:	Manager requests to create a new user story.
Response:	System queries Manager for the user story text and the sprint to which the story is assigned.
Stimulus:	Manager requests to change user story.
Response:	System queries Manager for the updated user story information.

3.4.3 Functional Requirements

- Story.Create: The system shall let an authenticated Manager create a new User Story.
- Story.Create.SetText: The system shall prompt the Manager for the text of the story.
- Story.Create.AssignSprint: The system shall prompt the Manager to assign the story to a sprint or to the backlog.
- Story.Create.Save: The system shall let the Manager save the changes to the new User Story.
- Story.Create.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.

- Story.Update: The system shall let an authenticated Manager update an existing User Story.
- Story.Update.SetText: The system shall prompt the Manager for the new text of the story.
- Story.Update.AssignSprint: The system shall prompt the Manager to assign the story to a new sprint or return it to the backlog.
- Story.Update.Save: The system shall let the Manager save the changes to the User Story.
- Story.Update.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Manager of the failure. The system will save the details and allow the Manager to retry the transaction.

3.5 Add Task

3.5.1 Description and Priority

An Employee or Manager whose identity has been verified may create a new task for a user story. Priority=High.

3.5.2 Stimulus/Response Sequences

Stimulus:	Employee requests to create a new task.
Response:	System queries the Employee for the task text and the user story to which it belongs.
Stimulus:	Employee requests to change task.
Response:	System queries Manager for the updated task information.

3.5.3 Functional Requirements

- Task.Create: The system shall let an authenticated Employee create a new Task.
- Task.Create.SetText: The system shall prompt the Employee for the text of the task.
- Task.Create.AssignStory: The system shall prompt the Employee to assign the task to a user story.
- Task.Create.Save: The system shall let the Employee save the changes to the new task.
- Task.Create.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Employee of the failure. The system will save the details and allow the Employee to retry the transaction.
- Task.Update: The system shall let an authenticated Employee update an existing task.
- Task.Update.SetText: The system shall prompt the Employee for the new text of the task.

- Task.Update.AssignStory: The system shall prompt the Employee to assign the task to a user story.
- Task.Update.Save: The system shall let the Employee save the changes to the task.
- Task.Update.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Employee of the failure. The system will save the details and allow the Employee to retry the transaction.

3.6 Assign Task

3.6.1 Description and Priority

An Employee or Manager whose identity has been verified may assign a task to an Employee. Priority=Medium.

3.6.2 Stimulus/Response Sequences

Stimulus:	Employee requests to assign a task.
Response:	System queries the Employee for the Employee to whom to assign the task.

3.6.3 Functional Requirements

- Task.Assign: The system shall let an authenticated Employee assign a task to an Employee.
- Task.Assign.ChooseEmployee: The system shall prompt the Employee for the Employee to whom to assign the task.
- Task.Assign.Save: The system shall let the Employee save the task assignment.
- Task.Assign.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Employee of the failure. The system will save the details and allow the Employee to retry the transaction.

3.7 Change Task Status

3.7.1 Description and Priority

An Employee or Manager whose identity has been verified may change the status of a task. Priority=Medium.

3.7.2 Stimulus/Response Sequences

Stimulus:	Employee requests to change the status of a task.
Response:	System queries the Employee for the new status of the task.

3.7.3 Functional Requirements

- Task.ChangeStatus: The system shall let an authenticated Employee change the status of a task.
- Task.ChangeStatus.ChooseStatus: The system shall prompt the Employee for the new status of the task.
- Task.ChangeStatus.Save: The system shall let the Employee save the task status change.
- Task.ChangeStatus.Failure: If any of the previous steps fails, the system shall roll back the transaction and notify the Employee of the failure. The system will save the details and allow the Employee to retry the transaction.

3.8 Visualize Sprint Burndown

3.8.1 Description and Priority

An Employee or Manager whose identity has been verified may view the burndown for a sprint. Priority=Medium.

3.8.2 Stimulus/Response Sequences

Stimulus:	Employee requests to view the burndown for a sprint.
Response:	System displays the burndown to the Employee in table form.

3.8.3 Functional Requirements

- Sprint.Burndown: The system shall let an authenticated Employee view the burndown for a sprint.

4 External Interface Requirements

4.1 User Interfaces

UI-1:	The system's UI shall conform to the *Windows User Experience Interaction Guidelines*[2].
UI-2:	The application shall permit users to use system with only a keyboard in addition to a keyboard and mouse combination.

4.2 Hardware Interfaces

No hardware interfaces have been identified.

4.3 Software Interfaces

SI-1: Active Directory
SI-1.1 The SPMS will request user credidentials from AD when started by the user.
SI-1.2 If AD authentication is unavailable or disabled, the SPMS will ask the user to authenticate themselves.

4.4 Communications Interfaces

CI-1: The SCRUM Project Management System will send an e-mail message to the team manager if a sprint's end date is reached and not all tasks are completed
CI-2: The SPMS will allow users to configure an e-mail message to be sent when the status of a task changes.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1 The system shall accommodate 9500 users during the workday of 8:00am to 6:00pm.
PE-2 Responses to queries shall take no longer than 10 seconds to be returned to the user.
PE-3 The system shall display confimation messges to users withing 5 seconds after the user submits information to the system.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

SE-1 Users shall be required to log in to the SPMS or be authenticated by Active Directory for all operations.
SE-2 The system shall permit only managers to create or edit projects, sprints, user stories, or teams in the system.
SE-3 Only users who have been authorized for remote access to the corporate Intranet may use the SPMS from non-company locations.
SE-4 Users may only edit tasks for projects that belong to teams of which they are a member.

5.4 Software Quality Attributes

Availability-1:	The SPMS shall be available to users on the corporate Intranet and to remote users 99.9% of the time between 7:00am and 7:00pm and 95% of the time between 7:00pm and 7:00am.
Robustness-1:	If the connection between the user and the system is broken while the user is editing data, the SPMS shall enable the user to recover the incomplete data.

6 Appendix A: Data Model

- User = An Employee or Manager
- User ID = A unique integer identifying a user
- Team = A group of Users with one User as the team lead and one user as the manager
- Team ID = A unique integer identifying a team
- Team Lead = A User responsible for leading a single team
- Team Manager = A User responsible for managing multiple teams
- Project = A software product to be produced using the SCRUM methodology
- Project ID = A unique sequential integer identifying a project
- Project Team = The team responsible for developing this project.
- Project Owner = A User who is ultimately responsible for this project. This is usually the Manager of the Team responsible for this project.
- Sprint = A one to three week period of development at the end of which the product is functional. One project is made up of many sprints
- Sprint ID = A unique sequential integer identifying a sprint
- User Story = A one-sentence description of a way in which a user would interact with the system under development used to guide development. Many user stories are completed during one Sprint.
- Story ID = A unique sequential integer identifying a user story.
- Task = A development task that would take one person six to eight hours to complete. One user story is made up of multiple tasks.
- Task ID = A unique sequential integer identifying a task.

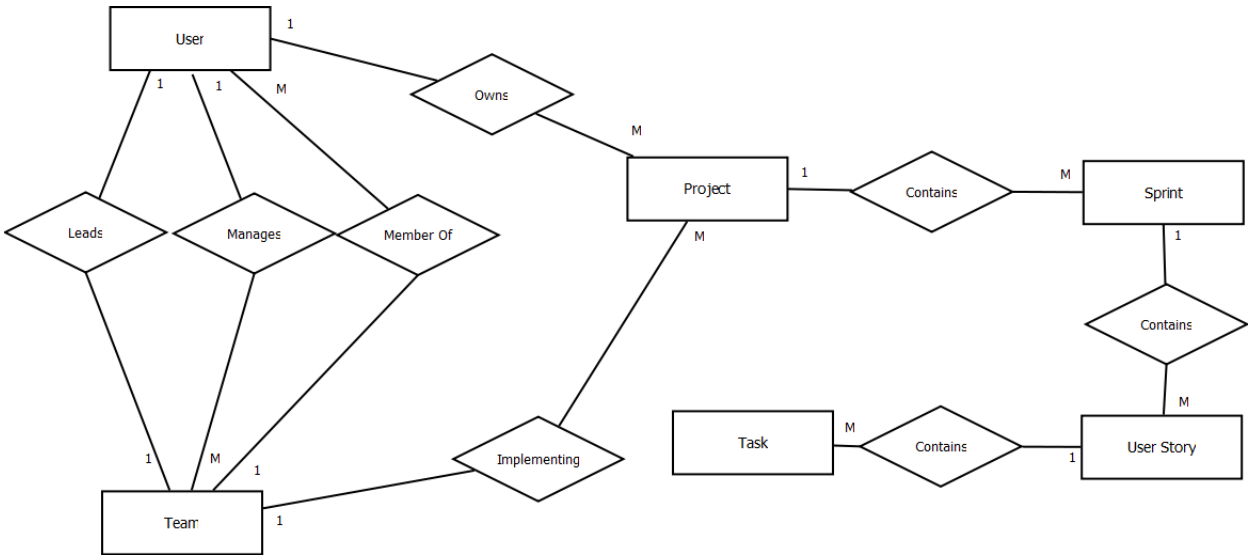

Figure 2: Data Model for Release 1.0 of the SCRUM Project Management System