User Story Management (1)

A class project for
CS 581: Advanced Software Engineering

Submitted by:
Essa Imhed
Hitesh N. Sharma
Sultan Alharthi
Ruth Torres Castillo
Vijaya Pandey

Submitted to:
Dr. Jonathan Cook

Date:
2/26/2015

Contents

1. Introduction

Jupiter is a user story management tool that allows users create and manage user stories. It also facilitates the user to generate requirements documents making a connection between user stories and requirement documents. It further allows users to make connections between user stories and the actual code.

Jupiter will provide an interface that gives the user the capability to enter his stories, then the system would generate a format of requirement documents based on entered stories, and finally to link these requirement documents to the actual code. For Example, a user story, U1, might be "As a user I want to have a comment box in the blog post so that I can put comments in the blog post." Based on the user story and the category the user story is assigned to, system will generate a requirement document that summarizes the user story along with other similar user stories.

As far as the management of user stories is concerned, user stories can be created and visualized in different ways. User stories can have different fields such as priority, assigned to, estimated time, tags, status etc and based on these categories, the system will provide different ways to visualize and arrange user stories. Further, if we get enough time, the system might also allow users to bulk load user stories or export user stories in a specified format for other user story management system to use.

The system will also have the facility of linking user stories to actual code by allowing users to specify particular line numbers from a particular file (which contains code) to be linked to one or more user stories.

Note: Further ideas to explore might be to create a flowchart to show the mapping from User story to requirement document to blocks of code in the file itself. We are yet to explore if wire-framing or class diagram design might also come in the flowchart. But we won’t have enough time to implement these ideas.

2. Process Outline

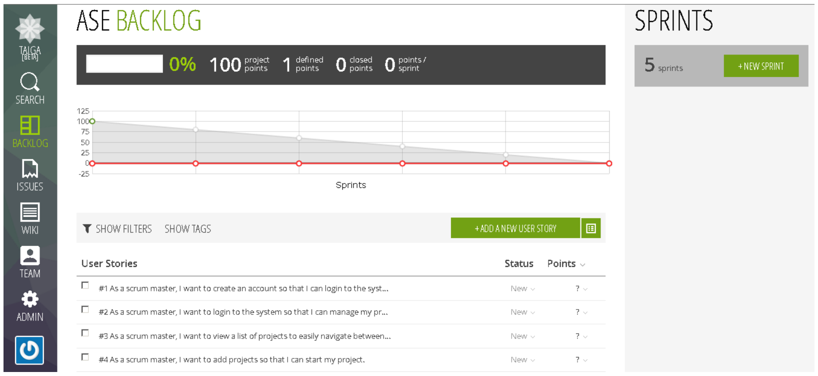
2.1 Process Description

We aim to follow the SCRUM process to manage this project. We have decided that we will have more than one scrum masters for the project i.e, each sprint will have a different scrum master. We will have a total of 5 sprints, each sprint spanning two weeks. The breakdown of Sprints and their timeline with designated Scrum Master is shown below.

Sprint	**Scrum Master**															
Sprint 1	Hitesh				■											
Sprint 2	Hitesh / Essa					■	■									
Sprint 3	Ruth							■	■							
Sprint 4	Sultan									■	■					
Sprint 5	Vijaya											■	■			
Project Ideation			░	░												
Documentation				▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓		
Testing							▒	▒	▒	▒	▒	▒	▒	▒		
		February				**March**				**April**				**May**		

Table 1: Sprint breakdown

We are currently using Taiga (www.taiga.io) open-source Agile project management system. Taiga is based on SCRUM and we plan to leverage its features to manage our project into sprints and also manage user stories of the project.



	User Stories	Status	Points
□	#1 As a scrum master, I want to create an account so that I can login to the syst...	New	?
□	#2 As a scrum master, I want to login to the system so that I can manage my pr...	New	?
□	#3 As a scrum master, I want to view a list of projects to easily navigate between...	New	?
□	#4 As a scrum master, I want to add projects so that I can start my project.	New	?

Fig: Taiga Project Management System

We have created some tentative user stories and inserted them into the backlog in Taiga. We plan to break down the user stories even more and assign it to different sprints.

Managing the number of story points for any particular sprint seems to be a challenging task. To start with, we will have 10 working days multiplied by the total number of project members (10 * 5 = 50) which gives us a total of 50 story points for each sprint. We are yet to decide the number of hours that will be equivalent to one story point, but we plan to start with 1 hour and maybe gradually increase it to 1.5 hours or 2 hours.

Table 2: Storypoint management

Story point Scheme	**Individual work hours per Sprint**	**Total Group hours per Sprint**
1 story point = 1 hour	10	10 * 5 = 50
1 story point = 1.5 hours	15	15 * 5 = 75
1 story point = 2 hours	20	20 * 5 = 100

Wunderlist application will be used for scheduling meetings to discuss about the project. Subversion (SVN) will be used as a software version and revision control system, where all documentation and code of this project is going to be posted.

2.2 Team Work Breakdown

We plan to have the following breakdown for work breakdown for individual team members.

Table 3: Task breakdown

Team Members **Tasks**	Essa	Hitesh	Ruth	Sultan	Vijaya
User Story Development	✓	✓	✓	✓	✓
Wireframes Development	✓	✓	✓	✓	✓

User Interface Design		✓		✓	
Testing, Test Scheme Development			✓		✓
Architecture Development	✓	✓		✓	
Database Design					✓
Coding and Implementation	✓	✓	✓	✓	✓

2.3 Feature Breakdown

The project aims to have the following main features:

F1. Add user stories: Users will be able to add user stories into the system. Each user story will have description, role, tag, priority, estimated time, assigned to.
F2. Interactive display of user stories: The user stories in the system will be displayed in an interactive way. Users will be able to view user stories based on priority, tags, assignee, time, status, completion etc.
F3. Link User story to actual code in the system: Users will be able to link user stories to particular blocks of code (line numbers in a file that contains the code).
F4. Generate requirements document from user story: User will be able to generate a certain format of requirements document form the user stories entered in the system.

F5. Bulk load user stories: Users will be able to add more than one user stories at a time. For example, if the user has written 20 user stories in a text file in a standard user story format, then the system can import all those user stories to the system. The format of user story might be “As a <user>, I want to …. so that ……”

Feature	Sprint	February				March				April			
	Sprint 1				■								
F1	Sprint 2					■	■						
F2	Sprint 3							■	■				
F3, F5	Sprint 4									■	■		
F4, F5	Sprint 5											■	■

Table 4: Feature Breakdown

2.4 Testing Scheme

Following Scrum technique the project will have user stories, every user story will have acceptance testing after they are developed. Once they passed the acceptance test every user story will be tested separately with unit testing in JUNIT.

2.5 Development Platform

The development of the project will be done in Java, based on client and server architecture. Java servlets and Tomcat server will facilitate the server side whereas the client side will be developed using Html, ajax, css and javascript. MySQL Server will be used as a backend database.