

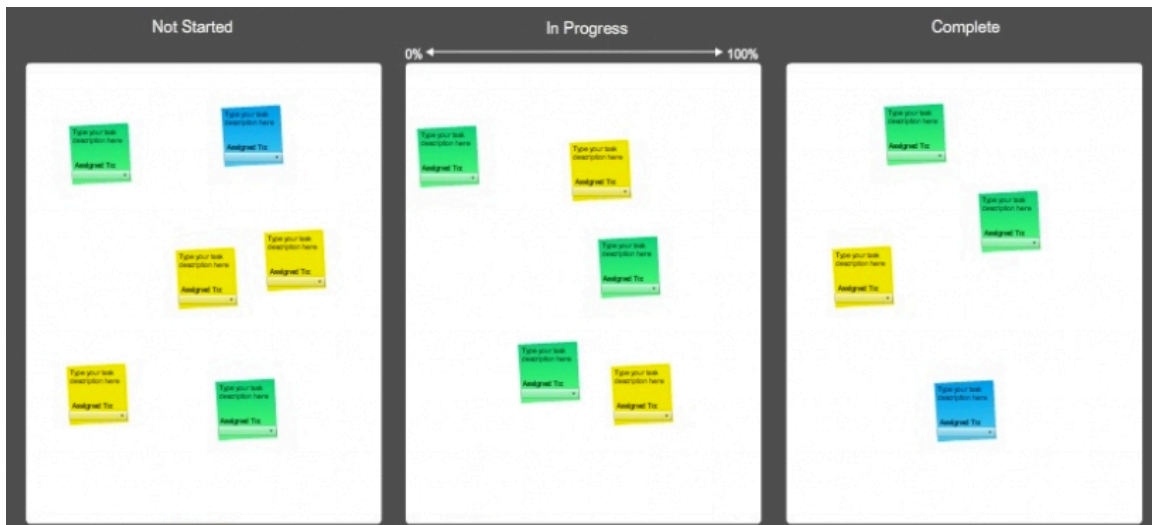
Virtual Sticky Board for Agile Project Management

Project Proposal for UCLA CS130

Introduction

Agile software development is one of the most used software development models at Cisco. To facilitate the Agile process, teams who are doing Agile development heavily rely on Rally—a project management tool for Agile development (<http://www.rallydev.com/>). While Rally is a very popular tool in the industry, it lacks certain aspects of Agile development. One of the most useful features that the teams need and that Rally fails to provide is the ability to visualize the detail progress of user stories(features). As a quick workaround, people write down their user stories on sticky notes and post them on physical corkboards. They draw lines on the corkboards to make columns that represent the different states of the process (eg. Backlog, In-progress, Done, Accepted, Published). This becomes a problem when the team members are spread across different geographical locations, since people cannot see the writing on the sticky notes clearly on cameras. Since it is a physical corkboard, it is not accessible to the team members all the time. In addition, there are some substates that user stories are put in on the corkboards, and Rally does not have such states. We need an application that will virtualize the process.

The project we would like to propose is a virtual sticky board project. Essentially, it will be a web application that will allow users to post stories in the corresponding state of the process, as in this example screenshot:



The data should be persistent and can be retrieved at anytime. The detail project requirements are listed in the project requirements section below. The

ultimate goal of the project is to integrate with Rally—getting data from Rally, changes made on this application will sync back with Rally—using Rally web services API. However, given the timeline of the class, we decided to split the project requirements into 2 sets. We expect the students to accomplish only the first set for this quarter.

Project Requirements

The first set

- ⇒ To provide persistent storage for all the following requirements, the app should integrate with its own database (via SOAP Web Services preferably).
- ⇒ Login for each user
- ⇒ 1 user belongs to multiple projects and 1 project will have multiple users
- ⇒ 1 project will have a workspace (User Stories progress view)
- ⇒ All team members of the project share the same workspace. Eg: A,B,C are the members of Project P. When either A,B or C logs in, they will see the same view.
- ⇒ User is able to add/edit/delete the User Stories in the workspace and save them in the database.
- ⇒ User should be able to categorize the User Stories into these default states:
 - * Backlog
 - Substates:
 - Product backlog
 - Release backlog
 - Sprint backlog
 - * In-Progress
 - Substates:
 - Requirement/Architecture
 - API review
 - Development/Implementation of features and tests,
 - Unit Test
 - Code review
 - QA
 - Documentation
 - * Completed
 - * Accepted
 - * Published
- ⇒ User should be able to categorize User Stories by color
- ⇒ Display the User Stories that the user entered on the corkboard
- ⇒ Display the User Stories under each corresponding state
- ⇒ The position of the User Stories roughly represents priority, and it should be persistent.
- ⇒ User should be able to move/drag around the sticky note from one state to another or from one position to another to change the priority.
- ⇒ The app should be compatible with Firefox.

⇒ Since this application will be ultimately integrated with Rally, it is important for the students to get familiar with Rally to align the database model with the states and data model in Rally. There is a 30-day trial version of Rally Enterprise edition that students can install. We can help them understand the data models on Rally.

The Second Set

Using web services API provided by Rally, the application will need to pull data from Rally and update the local storage when the user wants to. When the user wants to sync back, the app should write back to Rally keeping the states in sync, i.e., some adjustments need to be made to the mapping of the states since not all states from the app exist on Rally.

Proposed architecture

Database:	SQLite or MySQL
Language:	PHP or Python. (If the students know Python very well, they can choose Python. Otherwise, we recommend to use PHP)
Web server:	Apache
Development platform:	Linux/Mac or use a virtual machine on Windows
Approach:	Separate the workflow into Presentation (using CSS), User Interaction with the site (Javascript) and Functionality (Business logic). Get the functionality working first, then work on making things look nice.
Version control :	SVN: http://code.google.com/p/support/wiki/SubversionFAQ

Prerequisite

At least one student in the group must have database knowledge equivalent of CS143.

Incentives for the students

After the first set of requirements has been accomplished, students will be able to publish their work as an open-source project if they want to, since the project will be self-contained. After the second set is completed, the project can certainly complement Rally and can be published as an app for Rally. The project

March 9, 2011

also fits the nature of the UCLA class very well since the class is about software engineering principles and the application they will be building is an application for Agile software development. They will definitely get exposed to how Agile principles are being used in the industry.

Contacts

Ei Darli Aung (edarli@cisco.com)

Jonathan Yang (jonyan@cisco.com)