

Title

Multi-platform Kanban Taskboard

Abstract

The aim of the project is to design and implement a functional prototype for visualizing and operating a multi-platform virtual "Kanban Taskboard". Kanban (see References) is an agile methodology for visualizing, managing and optimizing the flow of software requirements for a given application and for distributing the associated work among the development team. Kanban methodology should be used by the team itself for the *Kanban Taskboard* development. The project should focus on interfaces and algorithms (not on technologies).

Introduction

The aim of the project is to design and implement a functional prototype for visualizing and operating a multi-platform virtual "Kanban Taskboard". Kanban is an agile methodology for visualizing, managing and optimizing the flow of software requirements for a given application and for distributing the associated work among the development team.

The *Kanban Taskboard* is to be operated exclusively by a development team working for application maintenance. However other users should be notified on taskboard changes through notification mechanisms such as email subscription and RSS.

The *Kanban Taskboard* should be operated entirely through a web interface, and partially through other technologies. It should be possible to integrate it simply with some team tasks repository. The *Kanban Taskboard* should have reports with team performance indicators for development activities, and is expected to generate activity and work outcome reports.

The project should additionally focus on selecting and implementing tracking and team productivity metrics. Prototype usability is also an issue to consider carefully. During work evaluation special attention will be paid to how the group addressed such issues.

The Kanban methodology should be used by the team itself for the *Kanban Taskboard* development.

The project should focus on interfaces and algorithms (not on technologies), as well as on architecture design.

The following sections present the expected functional scenarios of use, an example and some more detailed requirements. However requirements elicitation activities are expected from the development team, in order to specify the system requirements to be developed. Such requirements specifications should be part of the project deliverables.

Application domain and scenarios

Context: software development teams using the Kanban mechanism to control the work. Given that ubiquity is one of the main goals of the project, the tool would best suit the requirements of a distributed team.

Motivation: Kanban is one of the fastest growing agile methodologies today. A virtual Kanban Taskboard that supports such methodology and that can be accessed through different technologies would be of great interest for teams implementing it.

Need: a system which enables:

1. Software development using Kanban.
2. Access the Kanban Taskboard from different locations using different technologies.
3. Integration with other task repositories.
4. Notification mechanisms such as email notifications and RSS.
5. Generation of activity and work outcome reports.

An example: A development team working in the maintenance of any software that uses Kanban to organize and distribute the work. They need a system that facilitates the implementation of such methodology, makes the taskboard available from different locations using different technologies, implements some mechanism to integrate with other task repositories, has notification mechanisms and, lastly, generates activity and work outcome reports for management and improvement purpose.

Project goals (requirements)

The following is a list of functional requirements or project goals:

1. Access to the Kanban Taskboard must be available through different technologies:
 - a. Web and Mobile access is required and must be developed.
 - b. Desktop, Firefox add-in, Eclipse plug-in, Visual Studio plug-in and others are optional.
2. The Kanban Taskboard should provide an UI that emulates graphically a taskboard with tasks written on post-its. Optionally a drag & drop mechanism to move tasks should be provided.
3. Support to versioning the taskboard and the ability to compare versions should be provided. Chronological replay of the taskboard changes is an optional requirement.
4. A reporting mechanism so that team productivity and status can be monitored is needed. At least 4 reports must be included.
5. Mechanisms to subscribe to task state change must be provided. At least mail notifications and RSS subscription should be considered.
6. Importing and exporting of the complete backlog and of a given version of the taskboard is needed. Excel format is recommended for such imports/exports.

Intended output of the process (process focus)

Taking in consideration that the main goal of the project is to develop a tool to support the Kanban methodology, using this agile approach is a must during the development process. Teams are required to document the development accordingly with this methodology.

Tools and standards (optional)

It is strongly recommended to use a versioning system, preferably open-source.

It would be desirable to have good unit-test coverage. It would be even better if the application is developed Test Driven Development style.

The application building and installation process must be very simple and well documented.

Interaction between stakeholder and developing teams

The number of teams that could select this project is 3. Ideally, the teams should develop the project during the same period (e.g., in the same semester). The only mechanism to interact is through email in English or Spanish to [score2011@Pragmaconsultores.com](mailto:score2011@pragmaconsultores.com). The emails subject should begin with [SCORE-PROJECT-#] where # is 1, 2 o 3. The number is the team identifier provided by organizers. Ideally, teams should hail from different continents (e.g., one from North America, one from South America - preferably from Argentina- and one from Europe). The expected duration is the involved university Course. Expectations will be adjusted to course duration. However, I suggest duration not shorter than 5 months and no longer than 8 months). In February I will not be available.

References

1. <http://www.limitedwipsociety.org>
2. <http://www.infoq.com/minibooks/kanban-scrum-minibook>
3. Kanban, Successful Evolutionary Change for Technology Organizations by David Anderson