Web Based Scrum Board

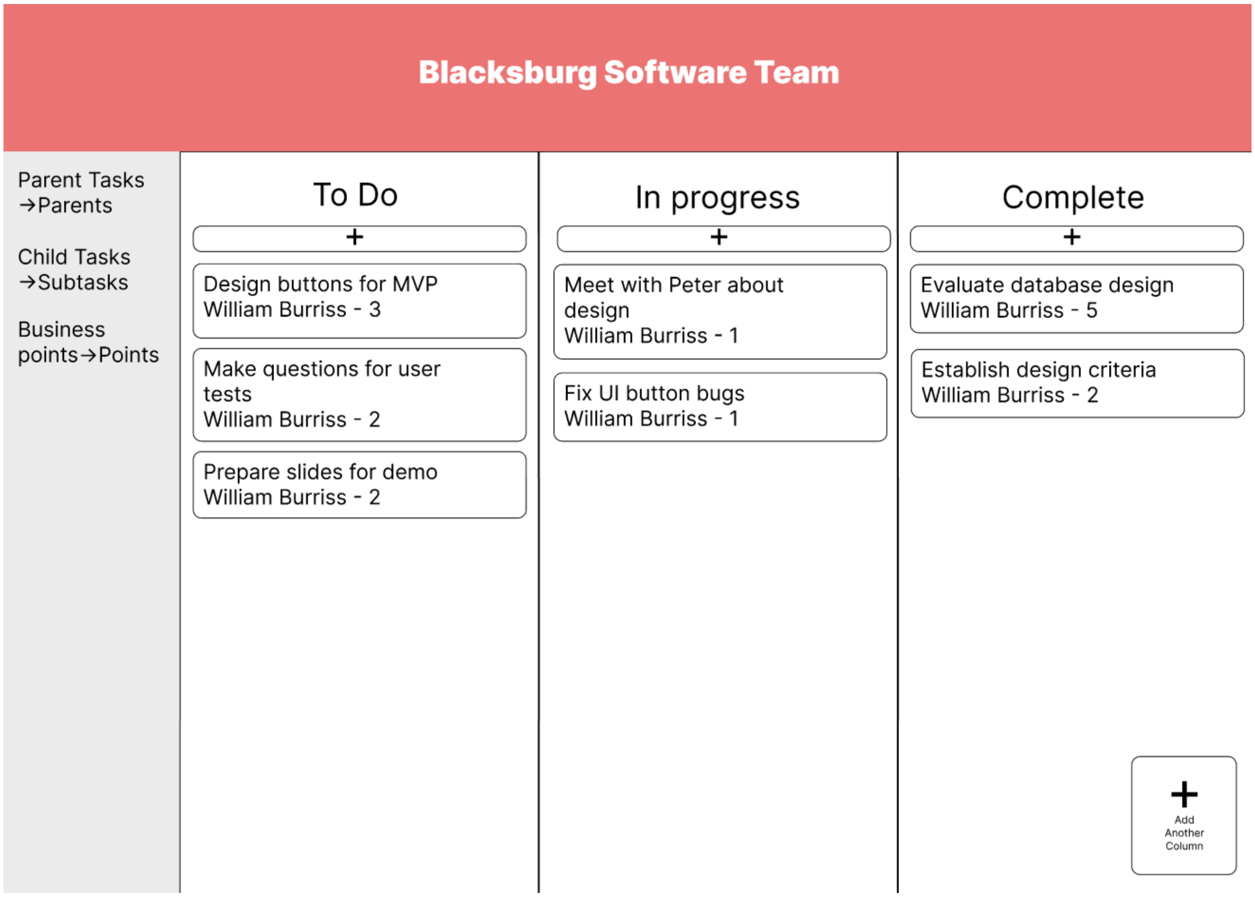
Project Milestone 2: Deliverable 1 and Requirements Analysis/Specification

CS 3704 Fall 2024 (83365)

Aditya Rao
Aidan Carraretto
Akshara Gandrakota
William Burriss

Process Deliverable I

For this milestone's deliverable we elected to construct a wireframe for our application. To recap, we will be constructing a web based scrum board that has many features we felt were lacking in commercial solutions.



Blacksburg Software Team

Parent Tasks →Parents

Child Tasks →Subtasks

Business points→Points

To Do	In progress	Complete
+	+	+
Design buttons for MVP William Burriss - 3	Meet with Peter about design William Burriss - 1	Evaluate database design William Burriss - 5
Make questions for user tests William Burriss - 2	Fix UI button bugs William Burriss - 1	Establish design criteria William Burriss - 2
Prepare slides for demo William Burriss - 2		

In the pictured wireframe we see many of the features discussed in the previous milestone. The ability to add columns is accomplished by clicking on the button in the bottom right corner. To add tasks you can click the buttons in each column. You can also move tasks between columns. On the left side is where you can change the names of epics and stories. In the wireframe above epics are called parents and stories are called subtasks. When constructing this prototype we considered some of the requirements we extracted from the interviews conducted with fellow students.

Requirements Analysis

Below is a list of 5 non-functional, 5 functional requirements, and 5 use cases for the scrum board system based on the results from our requirements elicitation. Overall, the requirements elicitation put into perspective how much users value agency and control over the organization of their projects and who can access and edit them, and our requirements reflect that.

Non-functional requirements:

Usability: Users must be allowed to import their own files, including images and videos, within tasks. This helps differentiate better between different tasks and can provide a more detailed summary or starting files of each task.

Reliability: The online scrum board service should only have downtime for updates, and should have about 7 days MTBF. This app shouldn't require frequent updates, so we can likely push for a relatively high MTBF.

Performance: Synchronous scrum board changes must be pushed to the database within 2 seconds on average. This goal is ideal for both quickly saving changes from users and for collaborative users quickly seeing any changes made by others.

Supportability: The system must not constrain users to a certain organization for their scrum boards; a great degree of freedom should be given to users who want to organize their project. In other words, the program should not limit the user to only a standard to-do -> in progress -> complete scrum board format.

Implementation/Constraints: Must be a web application, so will likely need to be made using Javascript. Any JS frameworks/libraries can be used, but no more than 3 should need to be used.

Functional requirements:

1. The web-based scrum board must provide the user with the choice to start with a scrum board template.
2. Users must be able to create any number of subtasks and provide descriptions for them under a column/parent of the user's choice.
3. When collaborating users refresh the page after another user makes a change, the change must be visible on their client (as stated by the performance requirement, within 2 seconds of the change being made).
4. Users should be able to sort subtasks by priority using a points system.

5. The scrum board must provide options for the user to customize the number of columns and the title of the column.

Use cases:

1. Create a scrum board

Precondition: User must have the website open.
Main flow: The user will click a button to create a new scrum board from the home page [S1]. The site provides the choice to use a standard scrum board template or create a board from scratch [S2]. User picks a template and is navigated to the board [S3].
Subflows:
[S1]: The user will click a button to create a new scrum board from the home page
[S2]: They are given the choice to use a standard scrum board template or create a board from scratch.
[S3]: User picks a template and is navigated to the board.

2. Invite a collaborator to an existing scrum board you own

Precondition: User must be on the page of a scrum board they own.
Main flow: User clicks a button to add a collaborator [S1]. Page prompts user to add collaborative users [S2]. User inputs the email address of the users they wish to add as collaborators [S3].
Subflows:
[S1]: User clicks a button to add a collaborator.
[S2]: Page prompts user to add collaborative users.
[S3]: User inputs the email address of the users they wish to add as collaborators.
Alternate flow:
[E1]: User changes the permissions of the collaborator to view only before adding them.

3. Create a column/parent task

Precondition: User must be on the page of an existing scrum board they have permissions to edit.
Main flow: User clicks the “Add another column” button in the corner of the screen [S1]. The page creates a new column on the right-most side of the board [S2]. The title of the column is selected for the user to change [S3].
Subflows:
[S1]: User clicks the “Add another column” button in the corner of the screen.
[S2]: The page creates a new column on the right-most side of the board.
[S3]: The title of the column is selected for the user to change.

4. Create a new subtask under a parent task.

Precondition: User must be on the page of an existing scrum board they have; the page must have at least one parent task.
Main flow: User clicks the + icon at the top of the parent task they wish to add a subtask to [S1]. The page inserts a blank subtask at the bottom of the column [S2]. The subtask is selected for the user to change the description and/or assign priority points [S3].
Subtasks:
[S1]: User clicks the + icon at the top of the parent task they wish to add a subtask to.
[S2]: The page inserts a blank subtask at the bottom of the column.
[S3]: The subtask is selected for the user to change the description and/or assign priority points.

5. Move a subtask between columns.

Precondition: User must be on the page of an existing scrum board they have; the page must have at least two parent tasks and one subtask.
Main flow: The user clicks and drags the subtask between columns [S1]. The page removes the subtask from the previous column and adds it to the column it was dragged into [S2].
Subflows:
[S1]: The user clicks and drags the subtask between columns.
[S2]: The page removes the subtask from the previous column and adds it to the column it was dragged into.

Requirements Specification

Below are four user stories for our application. The effort needed is calculated by assigning low complexity 2 function points, low to moderate complexity 3 function points, moderate complexity 4 function points, and high complexity 5 function points.

1.
User: I am a software developer and I want to be able to update my team on the status of my tasks (incomplete, in-progress, complete).
Acceptance criteria:

- The developer can choose status options to select for each task based on the admin’s choices.
- Status changes are updated in real time on the scrum board

- Team members are notified through notifications of their choice (email, app) when the status of a task has been changed.

Effort needed:

- The status selection feature is relatively simple, so it has 2 function points, the updating system is a high complexity, so it has 5 FP. Lastly, the notification system gets 3 function points. These 10 function points multiplied by 12 gives us 120 hours of effort.

2.

User: I am a project manager and I want to be able to assign tasks to my team members and direct project deadlines.

Acceptance criteria:

- The manager can input tasks and choose members to be responsible for them, then set a deadline for it.
- The person who was assigned the task will get a notification that they have to complete the task by a certain date.
- Tasks and the people assigned to them can be viewed through the dashboard.

Effort needed:

- The task assigning system is low to moderate complexity, so it gets 3 function points. The notification system is a little more complex, so it gets 4 function points. The dashboard interface also gets 4 function points. This totals to 12 function points, so 144 hours of effort is needed.

3.

User: I am a developer and I want to be able to add comments to documents I uploaded so my team members can have clarification on more tasks.

Acceptance criteria:

- The interface shows comments and allows people to be able to create and edit comments, as well as displaying timestamps.
- Only the allowed people have the permissions to view and add comments to certain tasks.
- Notifications are sent to the relevant parties as soon as the comments are posted.

Effort needed:

- The commenting interface is a higher complexity, so it gets 4 function points. Handling permissions is a little less complicated, so it has 3 function points. The notification system is also moderately complicated, so it also has 3 function points. This totals to 10 function points, so 120 hours of effort is required.

4.
User: I am an interested product owner and I want to be able to review the progress of the tasks going on, as well as give my feedback on certain parts.
Acceptance criteria:

- The product owner is able to view the status of certain tasks with details on each of them, but there are limitations on the information that will be visible.
- The owner is able to select a task, see the necessary details, and give feedback.
- Team members and the project manager are notified about any feedback, as well as the project manager gets notified about what is viewed.

Effort needed:

- The task dashboard with the modifications needed for the product owner to be able to access it is complex, so it gets 5 function points. Leaving feedback and slightly less complex, so it gets 3 function points. Lastly, the notification system is on the complex side, so it gets 5 function points. These 13 function points give 156 hours of effort.