ABB KANBAN BOARD

PROJECT REPORT

GROUP 1

Background

The project Digital Kanban board, a project suggestion from ABB Port. Christoffer Holmstedt, the team lead at the ABB Port, described the project suggestion and the reason behind the need of this project as follows.

ABB Port is a company that delivers automation and electrical systems for container and cargo handling. The company has several engineering teams that work on several projects. The problem for the teams has been a good way of visualizing the task for each team and even the process status for each task.
For solving the problem they have used a built-in kanban board from a software they are using called TeamForge. TeamForge is used for the projects, but the kanban board did not fulfill their requirements. The solution they came up with was to use a whiteboard for visualization of tasks, but it is not practical.

Digital kanban board project, as its name explains, is a kanban board as a web application that imports data from TeamForge. The board uses TeamForge login credentials, and, a local admin and user for situations when TeamForge is out of service.
The data that is being imported is what the people at ABB Ports call artifacts. Artifacts are basically an issue, something that needs to be done, and all of the accompanying information - a task in other words.

The benefits they want from this project is a system that every team member can access and use predefined functionalities. It will make it easier for all team members to track the tasks, and it is more practical to use for weekly meetings and even for presentation of the entire working process.

Deliverables

Deliverables we had during the project are project plan document, design description, product and the last one is the project report. For the deliverables, we have been done with them some day before the time required. The project report, final design description and a final product are the last deliverables during this course which should be done before the required time.

Deliverable	**Deadline**
Project plan document	November 16
First version of design description	November 30
First version of product document	November 30
Final version of product document	January 10
Final version of design description	January 10
Final version of project report	January 10

Table 1. Deliverables from project plan

Key Features

Our web application consists of the following features:

1. Login and logout
2. Kanban board preview
3. Cards drag and drop
4. Limit preview
5. Card preview
6. Creating/editing/removing parent category
7. Creating/editing/removing category
8. Creating/editing/removing swimlane
9. Selecting and importing projects from TeamForge REST API
10. Updating artifacts with TeamForge

1. Login and logout are available for both users and administrators. They are successfully logged in after using correct TeamForge credentials or credentials from the local database, in case TeamForge is down.

2. Once the users are logged in, they are able to view the current state of the Kanban board. It contains available categories (columns), swimlanes (rows) and belonging cards. Therefore, users are able to see what the team leaders find to be important and must be done.

3. Both user and administrator are able to drag and drop cards on the board and move them to the new state.

4. Each category consists of a limit which refers to a desirable number of cards for that column. If the limit is exceeded, label color turns to red as a warning.

5. To get more insight about the card, the user is able to click on it. Available information are: *id, artifact_id, category_id, swimlane_id, created_at, updated_at, assignedTo, description, title, category, lastModifyBy, teamId, closeDate, points, status*.

6. The administrator is able to create, edit or remove a parent category. Parent Category refers to a custom category which can contain several columns (categories).

7. The administrator is able to create a new category, and, edit or remove existing. New changes will be automatically updated with the Kanban board. For creating a new

category, the administrator has to fill in the following attributes: *name, limit, sort number, parent category*.

8. The administrator is able to create new swimlane, and, edit or remove existing one. New changes will be automatically updated with the Kanban board. For creating a new swimlane, the administrator has to fill in the following attributes: *sort number, name*.

9. Navigating to the *Project* page will allow the administrator to import artifacts from selected projects. Also, the administrator will be able to update existing projects in the local database with the new ones from TeamForge REST API. After selecting which projects should be included, the administrator is redirected to the *Artifacts* page.

10. In the *Artifacts* page, the administrator has the ability to select which artifacts should be shown on the Kanban board. In addition to this functionality, there is an option to update existing artifacts with the new ones from TeamForge REST API.

Results of acceptance testing

Our client had access to our software during the later stages of the development providing continuous feedback. So it was not necessary for the client to be involved in the acceptance testing. The acceptance testing was therefore performed by group members.

The user acceptance test was successful and all 24 tests passed. The full result of the acceptance test can be found in Appendix A. In the Appendix the URL can sometimes be presented with only the path. As an example https:kanbanboard.dev/login will be shortened to /login.

User Acceptance Test (UAT) Assumptions

- The domain name is kanbanboard.dev
- The domain secured
- TeamForge server accessible (when not specified otherwise)

UAT Entry Criteria

- Correctly configured
- Database seeded with example data using our database seeder

Missing Features or Requirements Not Met

We completed almost all the features or requirements planned for the system described in the project plan. Essentially, most of the things that were not met are simply improvements for the backend and features that could be improved in a certain way, then again, they can be considered as extensions instead of requirements since this doesn't affect the functionalities of the main product. The system indeed gather data from the ABB Ports TeamForge server, and update it in the interactive Kanban Board. The administrator should be able to configure the kanban-board rows and columns using navbar buttons representing the models as we wrote in the project plan, still there is one thing that couldn't be implemented in the project:

Filter artifacts

One of the features that was not met is the artifact filtering. This is similar to the filtering system for github issues (as seen in the figure 1.1). This functionality would allow the user to select and filter artifacts depending on their selected criteria, saving time for the user. This was planned in the beginning of the project. But the time was not enough to implement this feature.

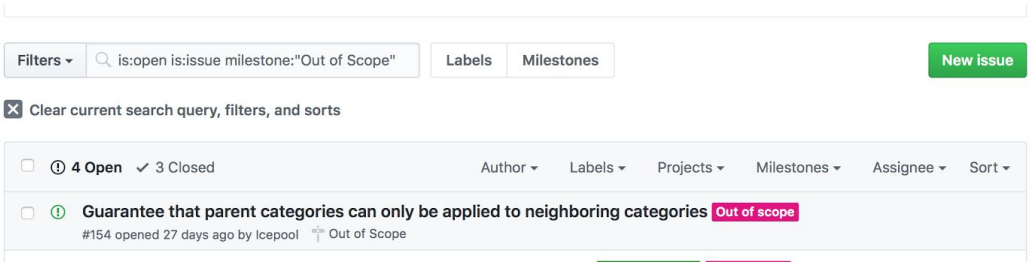


Figure 1. Filtering example from Github

Possible improvements and extensions to the product

These missing features would simply allow the team to make things that are already implemented smoother, which is a nice upgrade to the user experience. These possible improvement or extensions were planned as extras just in case the system was done with a lot of time before the deadline. Let's review the missing features:

Multi-Kanban board

The system developed is configurable for a single team. One of the extensions that can be added to the system is the multi-kanban board implementation. The client specified to develop a one team system. Our system may have the possibility to add different projects in the current system and also multiple users, but still there is only one Kanban board. It would be a nice extension to the product since this feature would allow multiple teams of ABB to use our system, but we should consider the scalability of the project regarding how many users would be connected at the same time.

Check for deleted Teamforge projects

Currently, the importation of TeamForge projects does not check for deletion of projects on the TeamForge side. This feature would allow to have a full synchronization with teamforge server. Nevertheless, since the beginning, the client specified that there is no problem having to do the deleting manually in this Kanban board system, so this feature becomes a possible improvement.

Identify and implement ways to improve board update functionality

The system has a functionality to handle the information of the board data after there is a board update request. So first, it reads the returned data from the API and then updates it in our Kanban board system with the help of AJAX and jQuery. There are better ways to do this thanks to Laravel framework, but the improvement is minor and is time consuming. Since the project is working with no problem, this makes this extension out of scope.

Upgrade authentication for API routes to use passport

Another of the improvements to the system is upgrading the authentication for API routes using *Laravel passport*. This would allow a full OAuth2 server implementation for the system. Currently authentication is done using standard web requests and session data. This isn't fully how an API should work. However, since there are no outward facing calls in the project (they are all internal calls made via AJAX calls), having this feature is not critical for the system or

even needed. This feature would just make the project to have an extra feature that is not needed.

Limit sort and not same number for categories/swimlanes

Currently, there are two blue arrow buttons in the *Categories* and *Swimlanes* panels. These buttons allow the user to easily move up or down the sort numbers of any category or swimlane with a simple click. However, there is a little bug in this, you can go to negative numbers if the down arrow is clicked when the sort number is 0. As well you can have the same sort numbers. This does not make the system to malfunction but requires the user to modify these mistakes manually in case they happen, which is not a design goal for any kind of project.

Guarantee that parent categories can only be applied to neighboring categories

Only categories that are beside each other in the Kanban board should share a parent category. For example, in the table 2., you should not be able to add a parent category to *Category1* and then to *Category3*. You should be forced to either move *Category3* to be beside *Category1* or add a parent category to *Category1* then to *Category2* then finally to *Category3*.

	Category1	**Category2**	**Category3**	**CategoryN**
Swimlane1				
...				
SwimlaneN				

Table 2. Fictional Kanban board

If a parent category is applied from *Category1* then *Category2* and then *Category3* but finally removed from *Category2*, the system should handle this. Otherwise we'll have two categories that aren't neighborssharing a parent category. If a parent category is applied to Category1 then to Category2 but one of the categories is moved, causing them to no longer be neighboring.

Project work

Changes to organization and routines

Most of the initial organization and planned routines have stayed the same through the project. Though there have been some changes. These includes the further development of some routines and use of established routines in other contexts.

An example of a routine that developed somewhat is code review. The initial plan was for it to be used during each completion of a feature to guarantee some code quality and early bug discovery. Once a feature was completed its issue was to be moved to “review” in the GitHub Kanban board and reviewed by another group member.

As development hadn’t started at the beginning of the project, it was used and continued to be used during the writing process. Both the project plan and design description document were reviewed using this system. This wasn’t the intent from the start but a natural byproduct of using the same system to keep track of both code and writing.

Another thing that stayed largely the same but developed somewhat was the use of GitHub issues. Upon discovering the usefulness of issue milestones for the project they were used to divide issues into weeks. This both aided in finding them at a later time and made some aspects of planning easier.

The milestones started out just dividing issues by week but once “out of scope” features and their related issues were defined then a milestone for those was created. This way all “out of scope” issues could be found in one place and be remembered.

At some point in the project issues had to be moved forward from one week to another. Once that happened there was a need to keep track of what week the issue had originally come from.

To do this issue labels were used. Previously labels like “Project Plan”, “Implementation” and “Design” were used. After this development week labels were also added to each issue. This way we were able to track if an issue had been moved. It also helped in identifying if more resources had to be put on a specific set of issues to aid in their completion.

All changes to routines and organization haven’t been developments. Some were the abandonment of something that wasn’t working and transition towards something that did.

An example of this was the brief usage of Facebook group chat before moving over to Slack.

The GitHub wiki was another tool that was used during the beginning of the project but slowly fell out of use. G Suite was simply more useful to the team and allowed for parallel editing and viewing.

In this same vein the GitHub wiki was also initially used for keeping track of notes taken after meetings. These notes weren’t very useful as they lead to many situations where the meaning

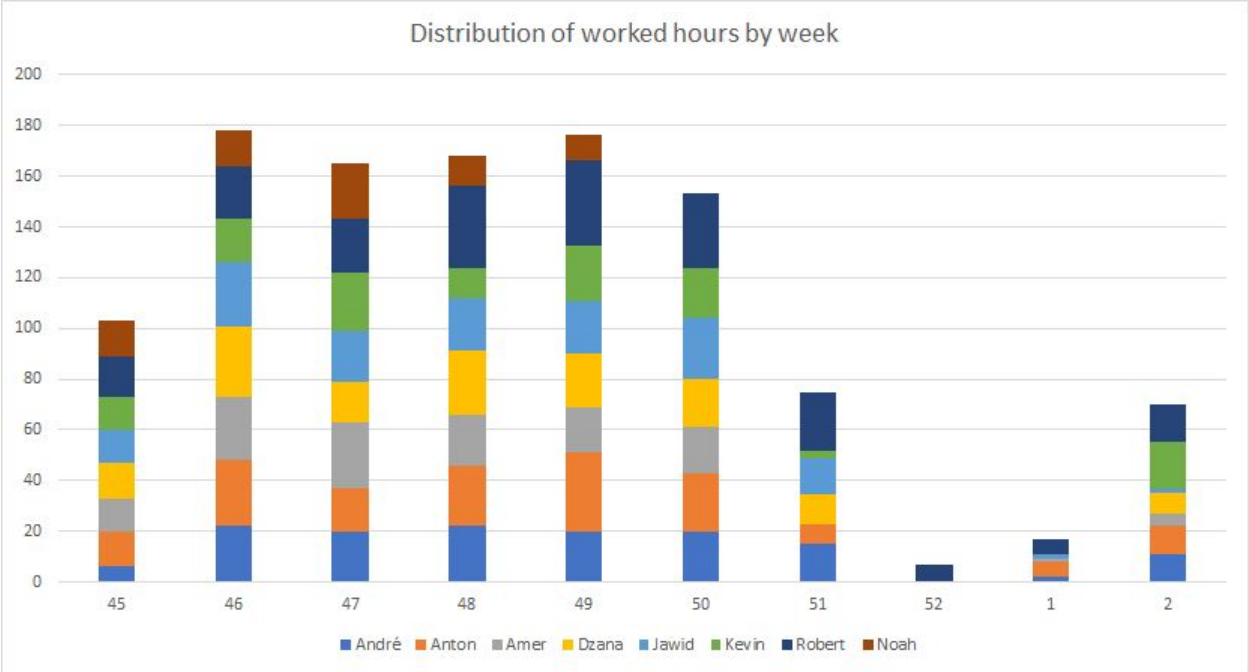
of the notes couldn’t be discerned later. Instead the group opted to making audio recordings of each meeting. This way the recordings could be listened to once something was unclear.

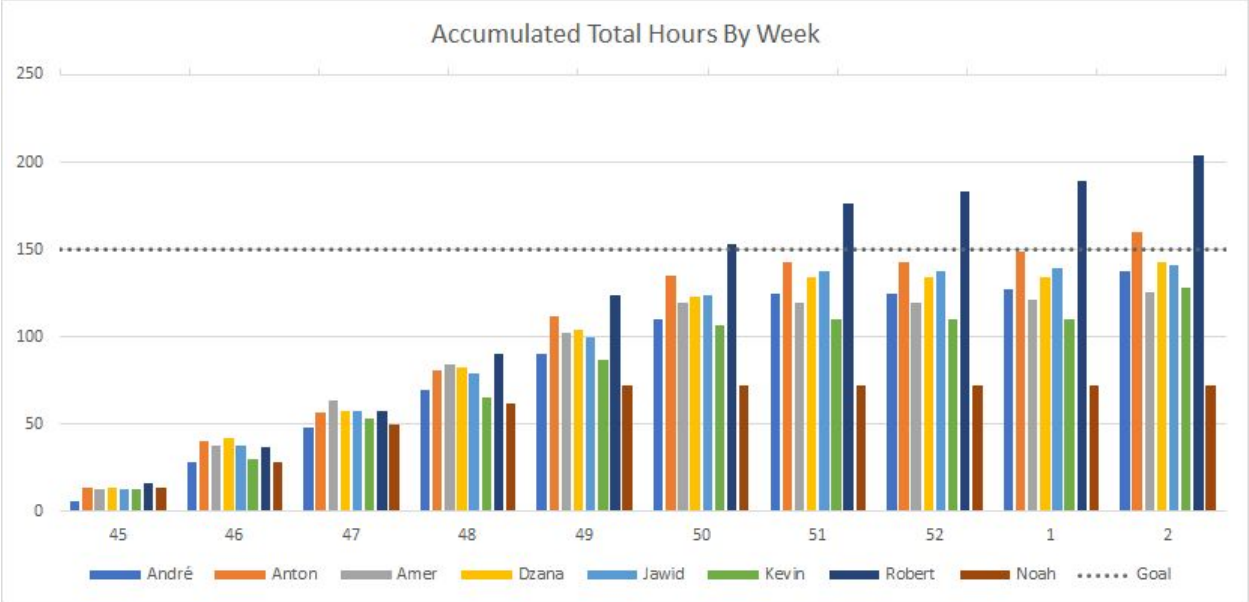
To make the most out of the information received from meetings the group also discussed the additional information after the meeting. This way a change of plans could be made while the conversation was fresh in everyone’s minds and issues could be defined.

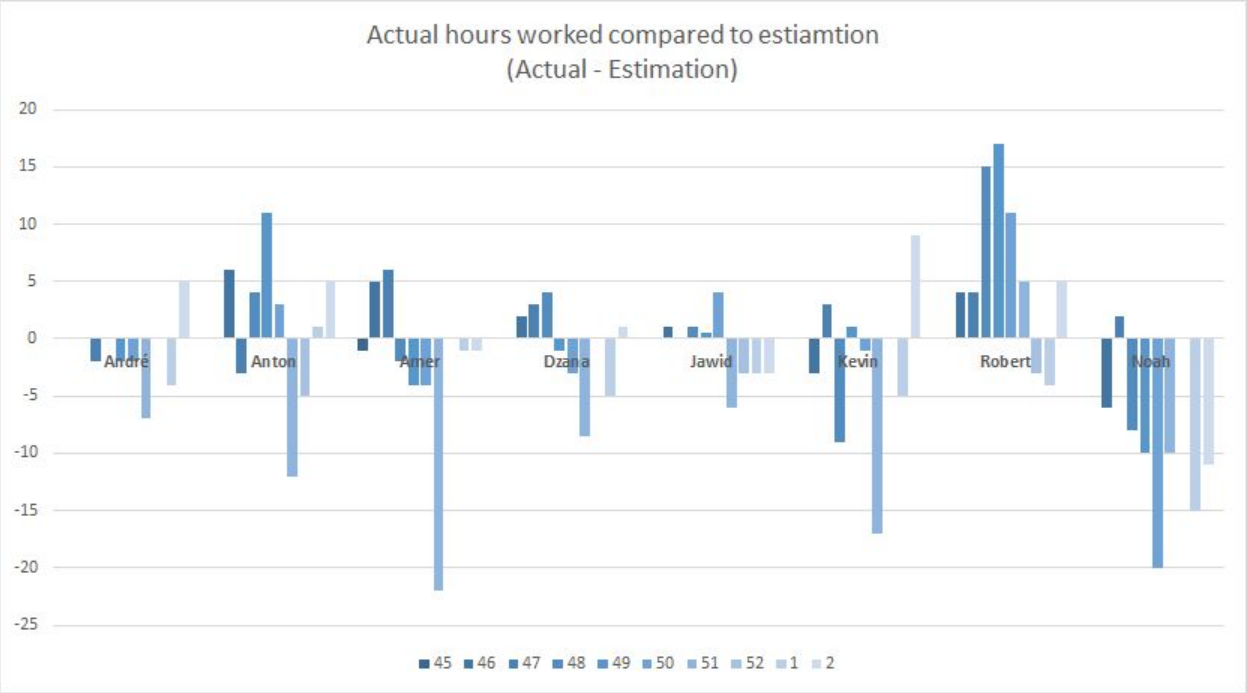
Finally, a thing that lead to some minor reorganization was the loss of a team member. With them dropping out their planned work had to be distributed among the rest of the group. But as they hadn’t done too much this wasn’t really a substantial change.

Total Project Effort

Worked hours per group member






Actual hours worked compared to estiamtion
(Actual - Estimation)
20
15
10
5
0
-5
-10
-15
-20
-25
André
Anton
Amer
Dzana
Jawid
Kevin
Robert
Noah
45 46 47 48 49 50 51 52 1 2


Time worked compared to 100% (150 hours)
160%
140%
120%
100%
80%
60%
40%
20%
0%
André
Anton
Amer
Dzana
Jawid
Kevin
Robert
Noah
%
% Goal

Distribution of work and responsibilities

Every friday before lunch we had a planning meeting. During that meeting every group member had a responsibility to help the group plan and set goals for the upcoming week. When the planning phase was done, the issues that needed to be solved for that week were distributed among the group members. The distribution of work did not really have any system behind it. It was mostly about making sure that everyone got about as much work as they wanted, and that not one single person had to take care of all of the boring parts. When a group member wanted something to work on, all they had to do was to have a look at the kanban board. They could then assign themselves to any issue that was not already taken.

Every monday for the last weeks we have had steering meetings. These meetings have been prepared and held by a pair of group members that are replaced every week. When someone had held a meeting, that person would not have to hold another meeting until everyone else in the group had held the same number of meetings. It was this pairs responsibility to look at the past week and se what the group had accomplished. They were also to compare those accomplishments with the goals that were set at the last steering meeting. Furthermore the pair should take note of what goals were set during the weekly friday planning meeting, and present those goals at the steering meeting. Lastly they had to summarise the total amount of hours worked by each and every group member, and present how those hours were spent.

The first presentation, the project plan presentation was planned and held by Anton Roslund, Jawid Nasiri, Kevin Oswaldo Cabrera Navarro and Robert Duras. They had the responsibility of making sure that everything was well thought out, and that they held a good presentation.

The second presentation, the design presentation, was planned and held by André Caldegren, Amer Surkovic, Dzana Hanic and Noah Gustavsson. They had the same responsibilities as the people holding the first presentation had.

The third and final presentation will be held by every group member except for Jawid Nasiri.

Some of the group members has had more specific roles. Anton Roslund has had the role of *Project Manager*, he has had the responsibility of communicating with our client, and doing management. Amer Surkovic has had responsibility over *Configuration Management*. Robert Duras has had the role *Wiki Editor/Organizer*, which has meant that he keeps track of our wiki. And finally Jawid Nasiri has had the role of *Secretary*, with the responsibility of secretarying, or in other word, write down what is said during our meetings.

Positive Experiences and Advices

What did we learn from the project?

The main focus of the course is teamwork. This involves planning, designing and implementing an end product. No matter the kind of project, you don't really need to have worked in a team before to know that it is hard and needs good communication and planning skills. Still we *re-learned* some things:

- Communication to the client/user is key:
    - Early client involvement is even better; it saves time because it removes lots of requirements modifications in the project latest's phases. Therefore, the team avoid to develop something that is going to be time, resource and effort wasted.
- Implementation of concepts of previous courses is the best way to learn.
    - This course is a perfect way to implement the concepts learned in theory-based courses like software engineer one, or also technical courses like web-development and databases.
- Little taste of how the industry works.
    - The interaction with a real client motivates the team to work, because in the end there is an end user waiting for a result. So this makes you work harder and more intelligent. This is an approach of how it works in the industry, so this project prepared us for bigger things in the future.
- Complementation between members.
    - There were members who know more about backend or Laravel, for other teammates designing the Kanban board was their main focus. So in the project we learned how to combine everyone's strengths in order to have the best final product we could develop.
    - In order to know everyone's skills, there should be a lot of communication all along the project, and a lot of honesty.
- Plan plan plan!
    - The most important thing in every software project is the planning. For us it was really well done, and the Fridays before the steering meetings were a nice way to measure the advance of the project and which direction to take the next step.
    - Some of the members would really love to code since the first day of the course, even me. But although small projects can succeed with some luck. Bigger projects require more planned approach.
    - Not to plan would have resulted in a massive mistake. It would have been chaotic in latest phases of the project, with ambiguous deadlines and not expected troubles.

- Some teams work better than others.
    - Some projects seem to be more interested than others but still all of them require to work together with the team
    - It's a big challenge to work with more people because usually the way of achieving tasks are very different from one to another.
    - This point is something you can see with both sides of a coin, either you find interesting to work with people you've never meet, or it could be a really hard time to deal with uninterested members.
- There is people with different motivations and goals.
    - The level of dedication and interest should be the same (or at least similar) in order to don't have problems with members falling behind.
    - Some members would be fine by just getting a passing grade, while others would not be okay without a five.
- Team manager is the key.
    - It is impossible to have a project heading to the right direction without a team manager in charge of it to control it. It must be somebody with great social skills and with a great sense of responsibility.

What were the problems?

- One member dropped the course, so the work needed to be reassigned.
- Holidays happened in the middle of the course, this made some members inactive during almost one to two weeks.
- Getting Teamforge was very time consuming.

What turned out as we expected? What will we improve the next time?

- Be to be aware of your capabilities, don't assign tasks that are not possible to do by yourself, and if you do not have any other option ask your teammates for help, maybe they know how to help you.
- Follow the deadlines that are in the course web page and follow this advice, ignoring the tips from people who had this course before is just ignoring the experience that we as a team got in this period.
- Try to complete as much as possible before Christmas. During the holidays the amount of work done is decreased significantly.
- Communication among team member might be the most important thing ever. If good communication doesn't exist people will have different ideas of what the team needs to achieve. So it's really important to sort out what view of the final product everyone has as soon as possible, so that everyone can work toward a common goal.