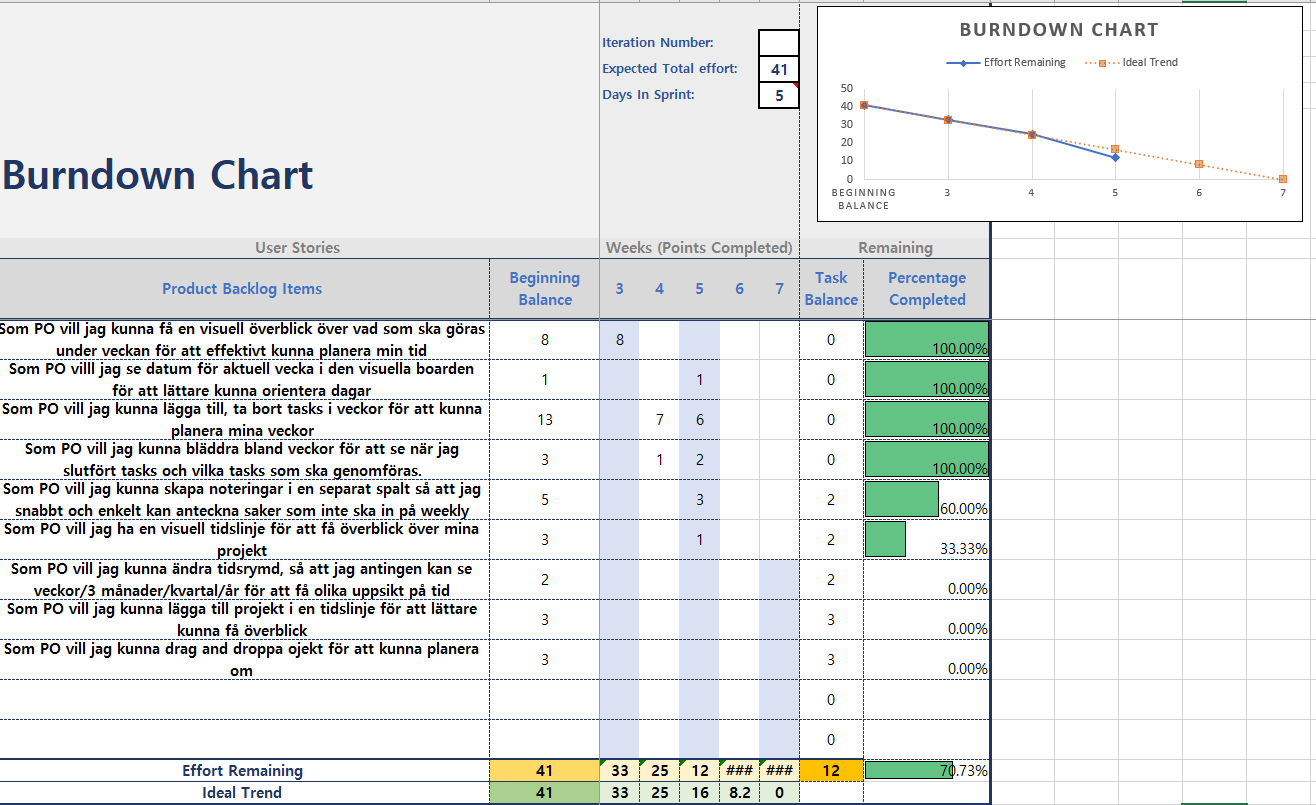
### **Customer Value and Scope**

* the chosen scope of the application under development including the priority of features and for whom you are creating value
  + This week the cosen scope continued in a similar way with the main goal of the finishing the MVP consisting of a interactive week-task-board and a porject timeline. Some small pivots have been done on the scope of the integration of the timeline with backend now that it is being developed in this sprint.
* the success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)
  + Finishing features. We have gotten close to finishing our week view and started working on the Timeline. These two features are our MVP.
  + Understanding how REST API is built is something that have become a larger part of what the backend people of the sprint team wants to achive. This because it is a standardized architcture that is used in many web applications and is very useful to know about.
* your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value
  + Added checklist for code review.
  + Implemented a clear list of what definition of done is for user stories.   
    It includes:
    - No regression of code tests
    - Code has been reviewed according to the code review checklist
    - The acceptance criteria for the specific user story has been met
    - The product owner considers the user story complete.
    - (We also try to keep our NFRs in mind)
  + Added a review column to the task board, user stories sits there when the feature has been implemented but has not yet been checked against definition of done.
* your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders
  + Added JUnit tests for backend to make sure no tasks disappear, this to make sure none of the costumer’s data gets lost during development.
  + Added UI testing for frontend to make sure that data that needs to be shown to the user is rendered to the page, this to prevent feature regressions. Which is desired by both the user and the developers.
* the three KPIs you use for monitoring your progress and how you use them to improve your process
  + Where we are now: We are using our 3 APIs below to monitor our progress. This week we had the following results.
    - Velocity 13



We also created a burndown chart to help planning the last sprints

* + - Customer satisfaction maxed out again..
      * Comment : Det känns som att funktionaliteten ökade från förra veckan och gött med timeline-vyn! Känns som att allt ni lägger till är relevant utifrån det vi pratat om och att hela projektet är på rätt väg
    - Code Coverage (22% lines)
  + Where we want to be: The coming week we want to get a better overview of where we are and the amount of work left in the project.
  + How we will get there: We will get there by creating a burndown chart for our application’s MVP

### **Social Contract and Effort**

* your [social contract (Links to an external site.)](https://www.scrum.org/resources/blog/how-make-social-contract-and-build-better-teams), i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives)
  + Added work on weekends, we finished this scrum on saturday because some people couldnt start until thursday.
* the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)
  + Add meeting time, and maybe solo hours. Currently we have not been tracking hours very accurately, but will try to fill in hours more frequently to see how the correlate with velocity.

### **Design decisions and product structure**

* how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value
  + Started using new forms of flex to get elements to function properly.
* which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)
  + Docs commenting for the Java files in backend, which in turn helps alot with making the code more understandable. This also enables quick generation of Java Docs.
  + API documentation in text document
  + Updated method names in frontend
* how you use and update your documentation throughout the sprints
  + Where we are now: We update API doc when new endpoints are added, use precise naming and comment logical methods. We consult the API doc when integrating the frontend with backend.
  + Where we want to be: We don’t feel like we need more documentation right now because it will take time away from creating value for our customer.
  + How we will get there: We will work with updating and maintaining the current documentation and focus on creating value in other ways.
* how you ensure code quality and enforce coding standards
  + Code review and checklistJunit ensures that the code works and that a good standard is used. The Junit testing makes sure that the code works properly and in the same time makes sure that code coverage is high, while code review helps with finding bugs before implementing the features to the feature branch.

### **Application of Scrum**

* the roles you have used within the team and their impact on your work

Just like previous weeks we have had a scrum master keeping our overall progress tracked and making sure we are keeping pace up.

However we have changed PO this week. previously our Customer was our PO, but to make we have a more smooth working flow one of our team members has taken over the role as PO. This member therefore makes sure that our customer is informed about what we are doing as well as forwarding his feedback to the rest of the team. Our PO also makes sure that the solution we develop always moves in the right direction.

The roles regarding frontend and backend Is keept just like before.

* the agile practices you have used and their impact on your work
  + We have updated the standard for the definition of done. This will make the coding easier because there is a checklist with everything that needs to be finished, in order for the code to be done. Therefore, there will be less code that is left broken and not finished that could be a problem in the future.
  + To ensure our code quality we have specified our code review with a checklist and code tests.
  + Moved merge requests to branches frontend and backend
  + For this week we moved the merge requests from master to the frontend and backend branches, because at the time we only do merges with these branches and therefor saw it fitting to change.
* the sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)
  + We uptated our cusomter on our progress by having a Zoom meeting and showing of our so far developed software. He was pleased with it but wanted us to focues I bit more on functionallity on the existing parts rather than starting with goals and metrics. He also expressed that drag and drop is a feautre he would like us to focus on. The feedback did not change our DoD, however it did change our coming userstories a bit.
* best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)
  + FlexFroggy, its a tool for learning how to use flex in css. This has been very helpful to develope deeper understanding regarding flexbox.
  + We have also started using Unit tests, which for some of us is a new tool that we use for our KPI code coverage.
  + We are also developing our usage of github, this greatly help our remote and individual workflow.