Danmarks Tekniske Universitet

02160 Agile Software Development:
Hospital Information Management Suite

26. marts 2019

OO Project plan - heisenbug

Authors:	***Study No:***
Andreas Heisenberg Engly	s170303
Anton Ruby Larsen	s174356
Karl Emil Takeuchi-Storm	s130377
Mads Esben Hansen	s174434
Mathias Jarl Jarby Søndergaard	s174426

Choice of Agile Framework

We will work with the SCRUM-framework, which will enable us to create a concrete schedule that can adapt if unforeseen issues arise. The timing on the sprints are based on a hourly rate, meaning we won't assign sprints to last for weeks, but "working hours" instead. In a SCRUM-framework the relationship between costumer and developer will often be central in the development of user stories and requirements. In our case, we will create the user stories ourselves from the way we have understood the requirements. The requirements will normally be specified in during negotiations between developer and customer. However, we do not have direct contact with the customer, and do not have the possibility of negotiations with them. The requirements will therefore be made through interpretation of the project and negotiations within the group.

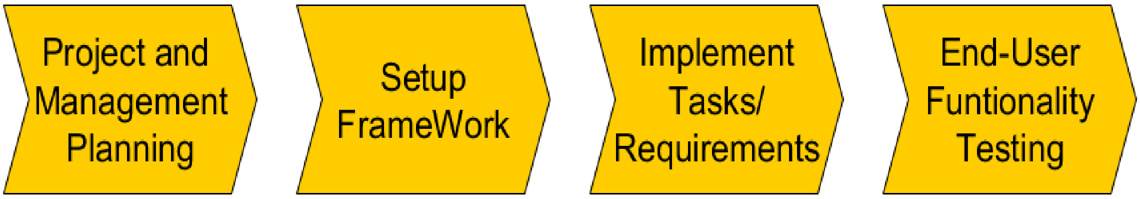
Project and Management Planning
Setup FrameWork
Implement Tasks/ Requirements
End-User Funtionality Testing

Figure 1: Overview for Project Flow

Backlog Management

Our backlog will initially contain all mandatory- and optional assignments. The mandatory functionalities constituting the back-end of the system will have the highest priority. Thereafter the front-end development will take place, since the system must run before a GUI is relevant. Excess time will be spend on implementing optional functionalities to provide a better user experience.

Task planning

All task planning is done at the "Retro Perspective" meeting. The main goal of these meetings is that everyone knows what to do and when it should be done. For instance going from requirements to scenarios is done through a group discussion, where everyone is allowed to make their points and tell their interpretation of the requirement. A "Retro Perspective" meeting will consist of four main elements.

1) Management process review.
2) Backlog re-prioritizing.
3) Re-factoring issues.
4) Sprint selection.

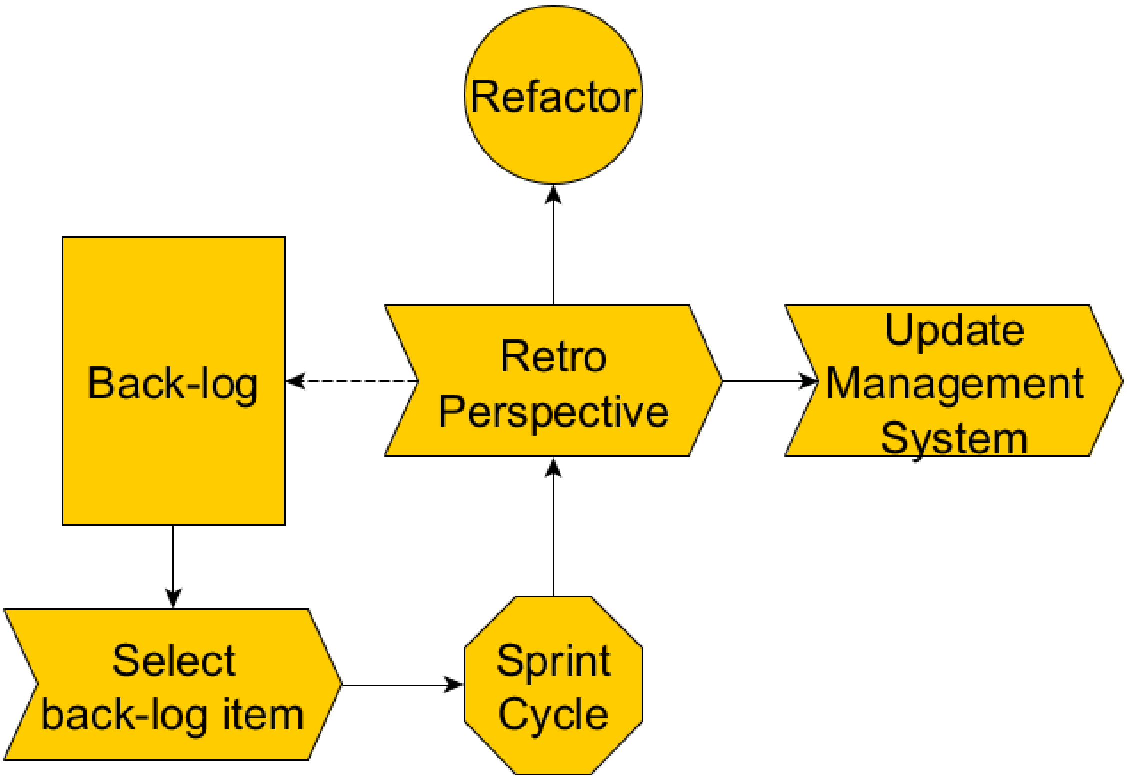

Figure 2: Flowchart for Management Process

Definition of Done

We define a SCRUM-sprint to be done when we have finish the following things:

1) User stories.
2) Cucumber tests.
3) JUnit tests covering as close to 100 percent of the module.
4) an UML diagram graphing the code.
5) The module (code).

6) User manuals (ReadMe).

Estimation and scheduling practices

To create a sustainable pedagogical schedule we will follow the SCRUM framework and assign a certain time frame for sprints. After every sprint we shall hold a meeting, discussing how things went. The key tool used is Trello, where a Kanban board is used for backlog management. Further details are described in "Best Practice"-appendix. Estimating the exact time amount for sprints will be hard. Thus we have discussed to be flexible with the time frame from sprint to sprint. If a sprint requires more or less time than planned, adjustments to our guiding schedule will be made.

Test strategy

At the start of sprint on a module we define a BDD-test from the user stories. Our main goal is to fulfill these initial tests. When these tests are completed, we investigate every part of the module by asking and answering as a minimum the following questions:

1) Are there any obsolete parts?
2) Has the goal of the module changed? If yes, does our module fulfill this new goal?
3) Can some of the parts be rewritten?
4) Is it implemented in a secure way (access modifiers, stuck state etc.) ?

These questions will be answered with the help of JUnit- and Cucumber-tests.

Team agreement

We have agreed on doing our best and to be time efficient. Good communication is crucial, especially if someone is delayed or can't attend a meeting. In order to be efficient we strive to create a friendly and helpful working environment where we can utilize each other's abilities. Being flexible is also a feature we should all adopt in order to overcome sudden changes and challenges that may occur doing a sprint. Finally, we should be respectful. If one finds themselves discriminated against, it will reduce their productivity by a great factor. At last it is expected that every member of the group follows the "Best Practice" policy.

What is Agile Mindset?

A loop based work process. We plan what to do in a defined time period and then work on those specified tasked in smaller teams in the defined time. When the time is up we meet and show what we have done and what to do now.

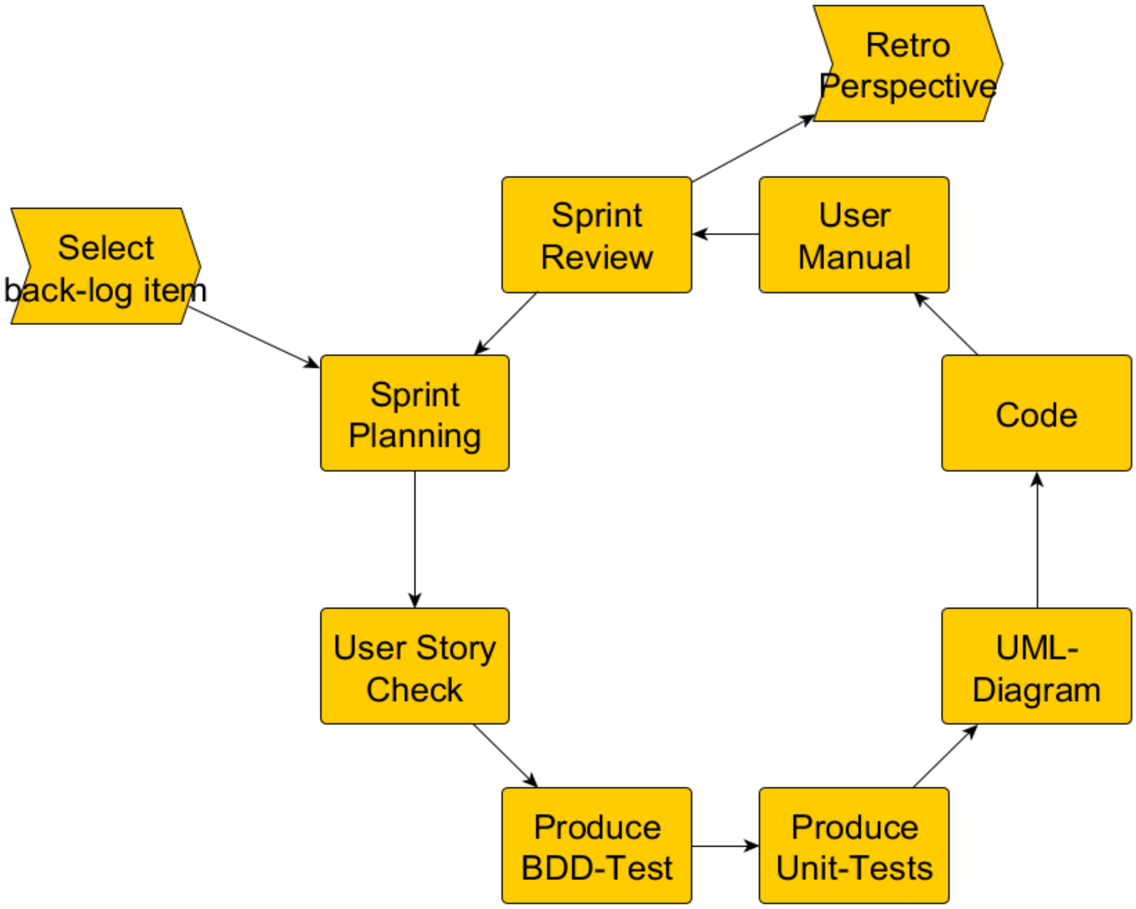

Figure 3: Flowchart of our adapted sprint rotation.

This structure is repeated and always evaluated up against an overall plan. If it wasn't in an educational setting the agile mindset would also include an ongoing customer contact. An example is user-stories. These would have been developed with the costumers to prevent that the final product does something completely different from what the costumer demands. We try to implement this by discussion internally in the group.

Metrics and Continuous Improvement Practices

As described earlier we will use a loop structure in our work process, where we continuously discuss how things are going. By doing this we are able to move people around to where we need more people to accomplish certain tasks. Furthermore, we are able to adjust the plan if we find unexpected challenges. In this way, no groups will get ahead of the rest. If this happens they can work without knowing if the others run into trouble. This can result in wasted work. Further both our management, agile- and internal structures must be updated continuesly.

Appendix A: Best Practice

Andreas Heisenberg Engly
Anton Ruby Larsen
Karl Emil Takeuchi-Storm
Mads Esben Hansen
Mathias Jarl Jarby Søndergaard

Management Tools: Trello

Whenever a group work on a new backlog item, the item should be moved to *progress* by dragging it from *backlog*. While working on the different items the checklist should continuously be updated such that the every member of the group knows how far we are with each aspect of the project.

The scheduled group meeting will be visible under *calender* in the upper right corner.

The tool can be accessed here *trello*.

Version-Control System: Git

Whenever working on new aspect of the backlog a new branch must be created. The branch should ideally be named [YOUR NAME] such that it is clear to everyone who the owner is.

Merging of branches are only allowed in such a way that they do not conflict with the work of any other subgroups. If in doubt of how merging will affect the existing work, keep the branch until the next group meeting.

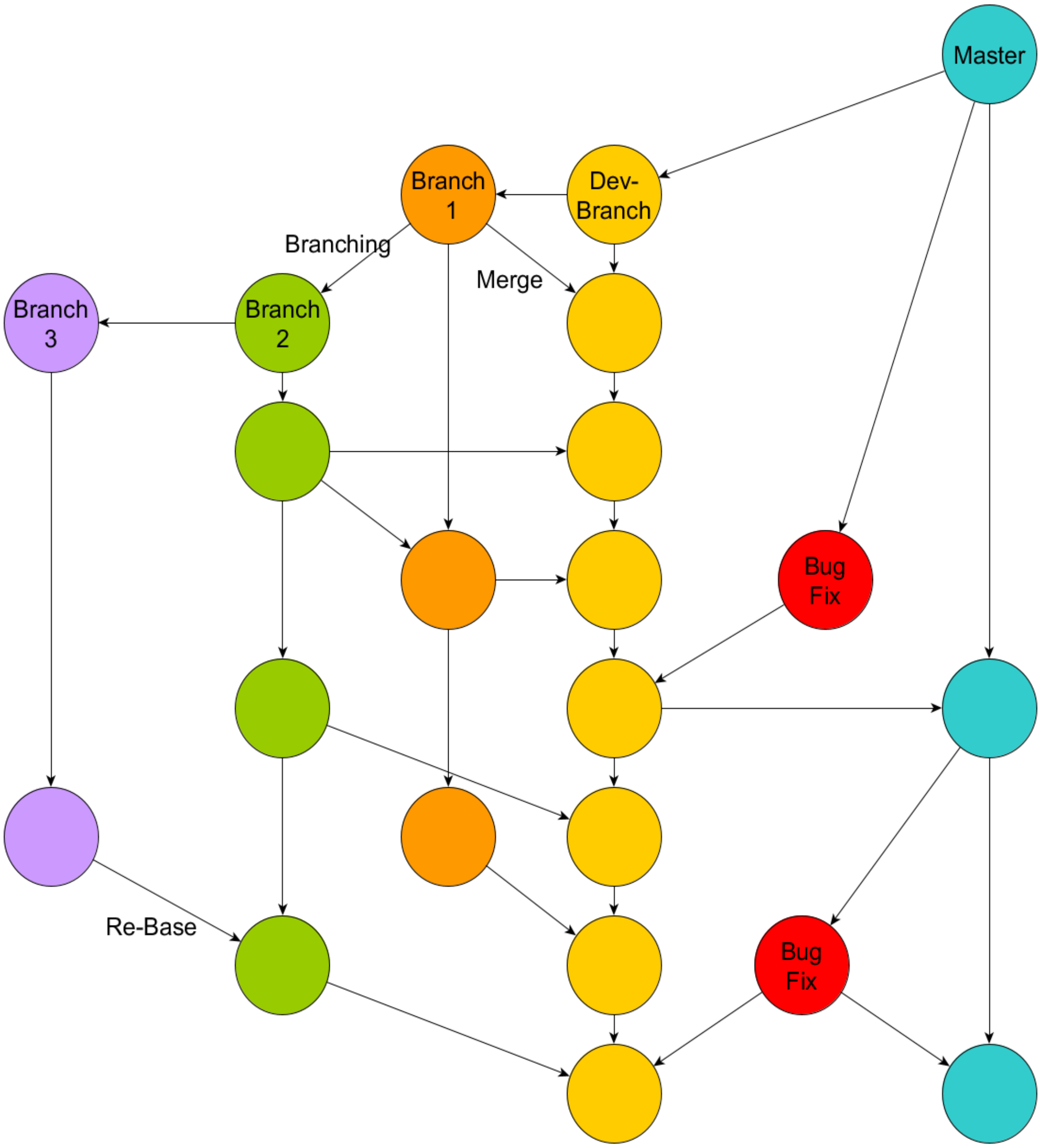

Figure 4: Workflow using Git(hub)

Communication: Telegram

The communication between group members will be maintained through Telegram. It will be used for questions, scheduling, and for personal matters of which the group members must be notified. Specifications for this practice will be reviewed at each group meeting.

Ongoing Report Writing: Overleaf

Under *Mandatory* and *Optional* on Overleaf folders for each functionality can be found.

For each functionality added by a subgroup a test-procedure must be described. The test description must contain a test-coverage in percentage, how it has been tested, and lastly the subgroup's thoughts on why the tests provided are necessary.
A UML-diagram must be created for each functionality. In that way it will be much easier to combine them into the final UML-diagram. It is recommended to use *Signavio*.
When a sub project is finished, a user manual must be provided. It must be clear for any users without previous knowledge of using IT-systems how to use the program to full extend.