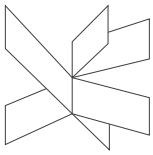


Process Report - ForecastMonitor

Tomas Izo, Georgi Stefanov, Ophelia Marott Zhang

Supervisor: Ole Ildsgaard Hougaard

Number of characters: 35368

ICT-Engineering

7th semester

12/10/18

Version: August, 2018
Template responsible: dans@via.dk

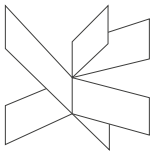


Table of content

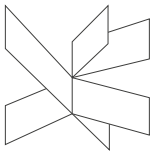
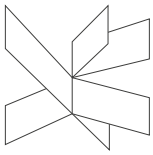
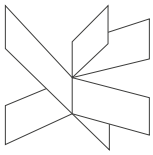


Table of figures



1 Introduction

This report seeks to document the entire development process of our bachelor project in respectively project initiation, description, and execution phases. The choices we have made, why we made them, and also how the practical circumstances have affected the process.



2 Group Description

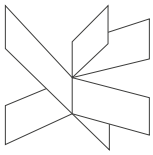
This section seeks to give an introduction of the project group members, their cultural background and project experience prior to the project start.

2.1 Georgi

Georgi comes from a small city in Bulgaria named Montana. Since the age of 15, Georgi has had a great passion for web development. In order to pursue his dream to become a web engineer, he left the city he grown up in, and came to Denmark to study ICT-engineering in VIA University College.
Georgi's specialization in VIA is Cross media, and he has worked on several projects besides his study in VIA, all in full stack web development. Despite his young age, he has rich experience in web development and usually acts as web specialist in his team, hence very often works alone. Georgi's preferred role in a group is the coder, so his favorite phase in a project's lifetime is the elaboration and construction phase.
Beside school, Georgi has one job as web developer, that requires him travelling to Herning once a week.

2.2 Tomas

Tomas is with his 29 life years, the oldest member of the group, he comes from Bratislava, the Capital city of Slovakia, but has been living in many places in Europe throughout his life.
Tomas' specialization in VIA University College is Embedded engineering, but he doesn't want to work with Embedded for the long run, instead many of his previous projects has been mostly C# and micro service (Windows Communication Foundation) correlated. Tomas prefers working in small groups and also like to be the coder in a project group.
Besides school, he has two jobs, both 15 hours per week. One as full-stack developer for DABAI, Systematic, where he works with python and AngularJS, another one as



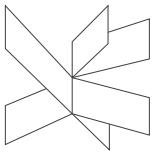
full-stack developer for OKIN Facility Prague, where he works with technologies as ASP.NET MVC, kendo, and knockout.

2.3 Ophelia

Ophelia lived first half of her life in China and the second half of her life in Denmark. She has since the beginning of her study been very interested in everything related to data engineering, and as a natural decision, she also choose Data Engineering as her specialization.

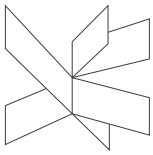
Ophelia has been quite around the world of software development, she has worked with full-stack development with ASP.NET MVC, administrated databases, MySQL and MongoDB, quality assured software with both manual and automated test designs, also developed larger systems from scratch with Python. Ophelia also prefers working in smaller groups, and she doesn't mind taking any role that is necessary for the team, but most often she ends up with the organizer role, which should take care of all the not so coding related work.

Beside school, Ophelia is doing freelancing as data specialist, without fixed hours per week and she also has a three-year-old child.



3 Project Initiation

This section seeks to explain how the project was initiated.
The group committed to the topic of this project together but based on different reasons.
Tomas was the original selector of the topic, and the most decisive one saying that he wants to work with machine learning in his bachelor project. Therefore, he actively contacted the lead developer of DABAI for potential bachelor project collaboration.
Ophelia agreed on the topic, because she has always been very interested in the concept of big data, pattern study in order to use historical data to help making future decisions. Furthermore, the topic is also very relevant for her specialization Data Engineering, which shares the enthusiasm about data with DABAI, which's primary objective is machine learning.
Georgi did not care about the topic of machine learning at all, though he is really interested in web technology, hence when we presented the opportunity of having an Angular front end, he agreed immediately.
The group was formed by Tomas one day asked Ophelia if she would be interested in joining him for bachelor project, where they will work with machine learning systems, and because of her interest in data, she said yes immediately. And then as Ophelia has previously worked with Georgi before, and was very impressed by his technical skill as an engineer, she asked Georgi to join as the last member.
We believe the planning of the project initiation phase went quite successfully, due to we knew Tomas will go on exchange half year in Prague in 6th semester, the group already had a handshake meeting before the start of 6th semester and BPR1 course, where the group together with the Product Owner and the lead developer of DABAI group, set up the main goal of this project, which is implementing a system for monitoring their current machine learning system for forecasting patient flow.
Furthermore, we also set some milestones for what we wanted to achieve before each document hand-in of BPR1 and arranged many skype meetings for further project scope and objective for this project, so we could have a successful project initiation phase, where the scope and outlines of the project is clarified and understood by all of the involved people.

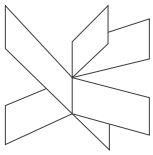


4 Project Description

This section seeks to explain the phase of project description, where the raw picture of project scope was drawn.

The group was aware of the importance of clarification of the problem before we could move on to engineering a solution product for the problem. Therefore, we arranged many clarification meetings, both with DABAI's lead engineer, so we could understand what their current problem is, and also with DABAI's user experience engineer, so we could have a better understanding of the problem domain. Hence the problem definition presented in the project description was written with close collaboration with the client of the system.

The group was aware of the limitation of time and resource we can put into the bachelor project as well, hence we knew we cannot have a ready for production product at the end of the project period, due to our choice of delimitation, where especially security and deployment of product to production server were put out of scope. Therefore, the overall goal for the bachelor project was to be building a monitoring system for the historical performance, where the features will be built onto it gradually with agile approach, so the most prioritized features will be done first, without requirement for an actual production ready result product at end of each agile iteration.



5 Project Execution

This section seeks to explain our implementation of the methods chosen for project management and execution:

- Feature driven development
- Scrum
- Early testing

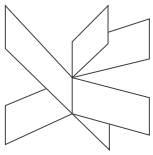
5.1 Feature driven development

Feature driven development is part of Systematic's standard software development process, and since two of three team members are already used to its way of working, it was a natural choice to take it in as the main project management approach. This provides several benefits, such as easier communication with Systematic and standardized documentation format.

With feature driven development, all the changes to a product must be planned, designed and implemented as features. For our case design by feature, and plan by feature meeting were held during the sprint 0 and 1, with feature list and feature roadmap as the resulting product. These products are open to changes throughout the whole project period, and we also did a formal review of them in the middle of the project period to make sure we still have a clear understanding of which direction the development should head to.

Each feature has a Feature lead, responsible for holding the compulsory FDD meetings and maintaining the feature documentation. Participants of the FDD meetings are everybody involved - developers as well as testers and stakeholders. In our case Ophelia has the role of a test engineering for all the features, and the only stakeholder from Systematic is our product owner, first Allan, then Adam.

The standard Systematic feature driven development approach has four compulsory meetings per feature: feature kick-off, ready for commitment, ready for implementation, and feature close. Based on the consideration of very small team size and only one involved stakeholder, the team didn't find the standard full-size feature driven development process necessary, hence feature kick-off meeting, which has main

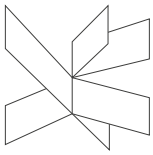


purpose for common understanding of the goal of the feature, and the feature close meeting, which is meant for everybody (developer, tester, and stakeholders) agreeing the feature is done, are not held, because all developers were involved in building feature list - hence all persist basic understanding of the feature goals, and feature kick-off can be held as part of Sprint Review, which is a part of our Scrum process already.
The ready for commitment meeting is therefore the first meeting to be held in the development process of one feature, it has the goal of clarifying the scope, assumptions, constrains, preconditions, requirements, acceptance criteria and risks of a feature (reviewing points 1 to 7 in the process overview below).
While the ready for implementation meeting has the purpose of every involved knows "how" the feature should be implemented technically, together with "how" the implementation should be validated (reviewing the rest of the points in the process overview below).

5.1.1 Feature Driven Development Process overview

The Systematic standard FDD documentation template contains following sections:

1. Feature Statement
2. Scope and Limitations
   - 2.1. In Scope
   - 2.2. Out of Scope
   - 2.3. Assumptions, constrains and preconditions
3. Estimate
4. Feature Requirements
   - 4.1. Requirements
   - 4.2. Acceptance Criteria
5. Risks
6. Dependencies
7. Clarifications and actions
8. User Interface Design
9. Technical Design



10. Deployment and operations
11. Test Approach
12. Stories and Other Activities

5.2 Scrum

Besides the feature driven development approach, scrum is used for managing the development process in the smaller scale. The entire project period was cut down into sprints of two weeks length. Each sprint was started with a sprint planning meeting, clarifying the focus of that sprint together with the product owner, and the scope of the sprint by defining the sprint backlog.

During the sprint, we have twice daily scrum a week (because each team member committed to work two days a week), where each team member addresses what he/she has been doing since last scrum meeting, what he/she is going to do until the next scrum meeting, and if there are any obstacles.

At the end of each sprint, we hold sprint review, where we demo the result of the sprint to the product owner, getting feedback if the development is on the right track, and we are actually developing a product that fulfills the need of the user. After the review, we also have sprint retrospective, where we discuss the process of last sprint, if there is anything good, we should keep doing or do more, and if there is anything that could be done better or could start doing.

For our case, we have also invited our supervisor for sprint review and planning, where we could get review for our work from the academic perspective and address eventual questions for how to carry on further work.

We did not have a standard product backlog, because our feature roadmap covers the same functionality of it.

We have assigned one person for being scrum master, who has the responsibility to document our work progress, call in to scrum ceremonies, initiating the sprint backlog for each sprint, making sure everybody has what they need in order to work optimally. Ophelia was assigned the scrum master in the beginning until she got sick, then Tomas took over.

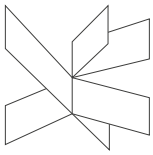


Figure below shows a burndown chart from sprint 6. The black line represents an ideal progress – bounded by its +/- 10% (dashed lines) and orange is the actual one. Tasks and stories not completed within a sprint would be carried over to the next one – these were usually stories waiting for verification or larger tasks spanning through multiple sprints.

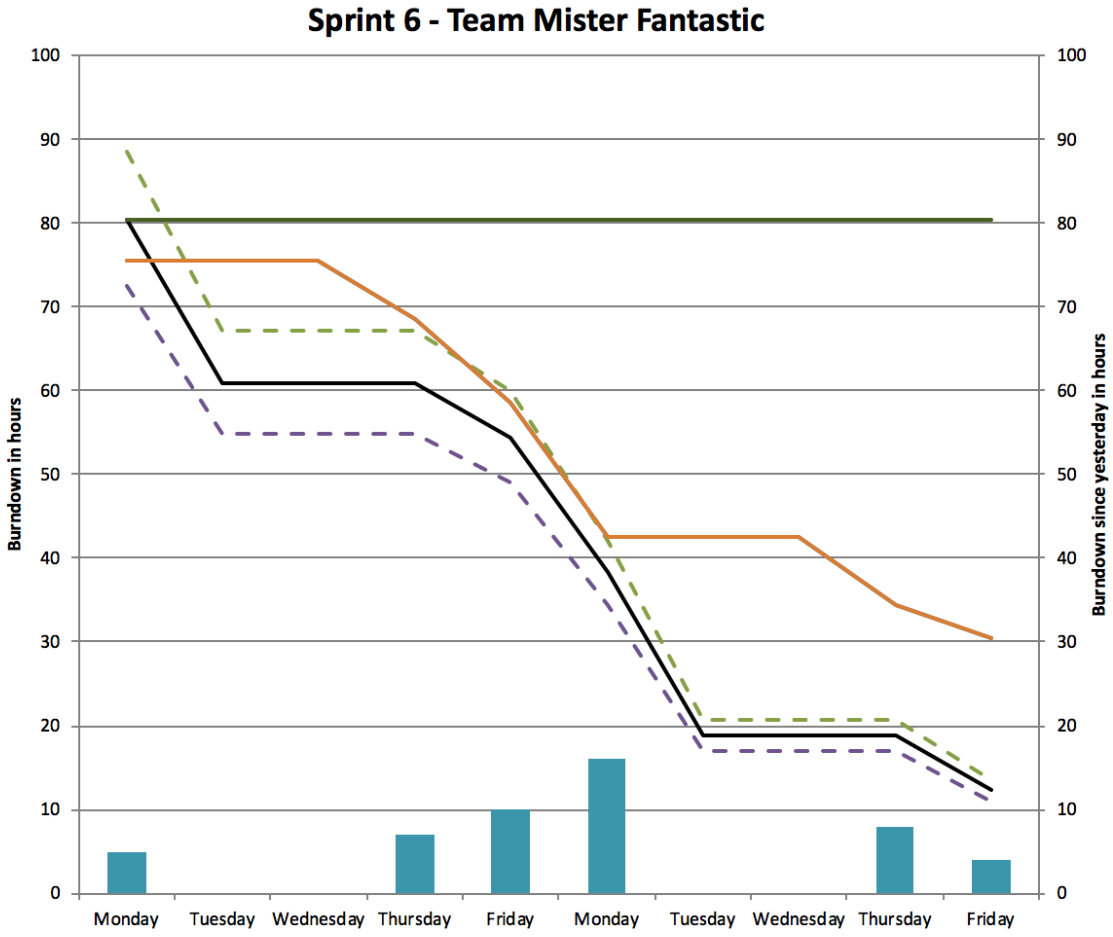
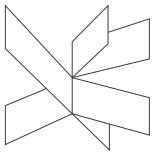


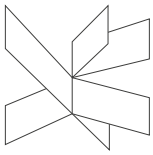
Figure 1 Sprint 6 - Burndown Chart

5.3 Early testing

Early testing is a strategy we have chosen to use as a part of the general quality assurance strategy in the project. We execute it by having the business logic tests as low level as possible implemented along with the actual system implementation, e.g. unit test preferred than GUI test.



This is done by, after the tester has defined the business logic test cases to be covered in a certain feature, she will have an informal talk with the responsible developer, where they will together deduct, which test level the test could be at lowest. As unit test is always the most preferred test level, she will always start asking if the business logic could be validated with unit test (unless it obviously cannot), if the developer denies the option because of technical reasons, the same question will be asked about a higher possible test level, until both parts agree on a certain test level.



6 Personal Reflections

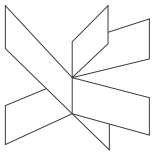
6.1 Ophelia

Already before the start of the project, because of my background in software testing (had an internship as test engineer and a certification) and my specialization in data engineering, my role and functionality in the team was predefined as the "data girl" and the "tester". And then at the beginning of the project, because I did not have a student job next to college as the others at the point, I also took the responsibility of Scrum Master on me. I liked the clear definition of responsibility, it made the working process easier.

The beginning of the project was a bit challenging, as all the team members and the Product Owner from Systematic has to get used to working with each other (we never really worked together before), and we also had to figure it out how much of the feature driven development and scrum process makes sense to be used for exactly our project. As Scrum Master as that time, I did my best by arranging meetings - all the recurrent scrum ceremonies as well as the meetings required to start up a feature driven project, and I also helped Georgi to start up with working on Systematic's platform by giving him tour of basic how to with the technologies used by Systematic - Jira, wiki, bitbucket.

In general, I believe everybody did their best, but we probably had different opinions about what to get out of this project. Tomas once told me, that he wanted to build as big a system as possible, while I really wanted to do an "academically correct" project, so I tried my best to push the project into that direction by putting more effort in documentations than what Systematic will normally require from their development process.

Around one month into the project, I got very ill. Since then the entire project period has been very tough time for me with many hospitalization trips, I wanted to give up at several points, because I did not feel like I could put as much effort into the project as I wanted. Luckily my group has shown a lot of understanding for my situation and allowed me to continue and contributing as much as I could, where we made a deal, that I will from then mainly act as the tester for the project while also taking the main



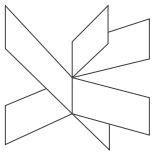
responsibility for report writing. Tomas took over my responsibility as Scrum Master, and both agreed on writing automatic tests themselves.
For me, the project has carried out as good as I could make it, and it is difficult to say what I could have done different, because of the practical circumstance. At the end I am eternal grateful for my team, without them and their understanding, I would not be able to finish this project.

6.2 Georgi

In the beginning of the project I was a bit insecure of how much of my skills will be in use for the project and how much I will be able to help. My strongest side is web development and from the looks of it in the beginning I was the “front end guy”, I had no knowledge in machine learning or feature driven development and I’ve never worked with Systematic before but that was only a reason for me to put even more efforts in the project.

When I first gained access to the computer Systematic provided, for me to work Ophelia was really helpful to “show me around” their system. She showed me how to use the wiki, Jira and also their bitbucket which helped me to get on track quick and start working on the project. It was difficult for me to contribute fully to the project in the beginning since most of the work was about feature driven development and all the required steps that needs to be done before we start coding, like RFC and RFI meetings. In the first two features (feature 0 and 1) I needed some help from the team to be able to do my part, so Tomas gave me some links where I can read about feature driven development and how Systematic is using it and Ophelia explained me the process in a one of our conversations. Soon after feature 1 was RFI-ed we decided that I can be a feature lead for the next feature since it was mostly about the front end. Being a feature lead for the first time was scary but it helped me gain confidence and better understand the process.

Somewhere in the middle of the project period Ophelia got really sick and we were unsure if she will be capable for finishing the project. Because of that with Tomas we decided to reprioritize the features and shuffle the roles in the team so at the end we



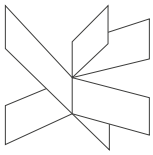
will at least deliver the most important features. Tomas took over the role of Scrum master and I was assigned as a feature lead for the final feature we decided to implement. It was a small feature but a necessary one and I was confident that this time it will be no problem for me to be the feature lead.

From the moment Ophelia got sick we had several conversations about her condition, and I was sure that she really wants to see the end of the project. During one of the meetings with our supervisor we brought up the situation with Ophelia, after discussing it we decided to give her another chance which she took and didn't let us down.

Overall, I learned a lot from this project, I realized that I can be helpful in other areas and not only coding. I saw how a fairly complicated system can be broken down to smaller parts, planned and implemented in an understandable and easy way by using feature driven development. I've also saw how a large company is working and communicating and delegating the work based on their roles.

6.3 Tomas

I got the idea of making an AI related project already during my internship and just before the end of it we assembled a team together with Ophelia and Georgi. I spent the next semester abroad and honestly, it was primarily them who were in touch with Systematic and prepared the whole project – and I must add that they did a really good job. Shortly after my arrival, I got a part-time job at the AI (DABAI) department, so I spent a lot of time on site and often acted as a mediator between the two parties. I was also the only one, who had access to the Forecast System source code and I'd make a change to it, when it was needed. We've had an early but a slow start, but were continuously gaining momentum, and if it was not for Ophelia's sickness, I'm fairly confident we'd have completed all the features. The backend (and test projects) required considerably more work than the UI and without her help, I often felt as a bottleneck. As the time progressed, I've had to take on more of her responsibilities, which would keep me quite busy…

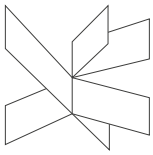


Taking all the circumstances into consideration, I'd say we did very well. Teamwork worked well, including supervision from both VIA and Systematic, and I particularly enjoyed implementing the system with Georgi. I was already used to the process previously and had no problems with following it, but reflecting on it within the current project, I feel that it may have not been the right one for 2 developers, working roughly 2-3 day a week.

On a personal note, I'm glad I got to gain some extra .Net Core experience and feel it'll help me professionally. However, I'm a little sorry that we did not manage to implement some extra features, because they looked quite interesting. Last, but not least, it feels really good to be done, as I put in many more hours in than I anticipated.

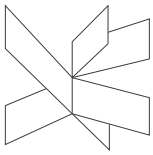
7 Supervisor

In general, the group is very satisfied about the result of how we interacted and used our supervisor. We invited our supervisor to our recurrent scrum end-and-start meetings, so the supervisor has an eye on our progress and working approach continuously, as we communicated with the supervisor at least once every second week.
We used our supervisor actively whenever we have meeting, we note down questions for him during the sprint, and ask him them during the meeting. If we need review for some report sections, we send the document during the sprint and ask for review at the next meeting. And our supervisor has always been very good and clear at answering our questions and giving feedback.
We used our supervisor mostly for help making a good academic project, so report content has been the main topic of our communication, though he also did cut in with his concern of our lack of communication in the beginning of the project, which helped us improve the group work afterwards.



8 Conclusions

Our choice of using Feature driven development together with Scrum has worked out well with Systematic's normal process and eased the communication with them. While the use of Early testing has also benefited our project with a structured strategy for quality assurance process. Overall, even despite the many practical obstacles during the process of our bachelor project, the group has ended up finding a harmonic solution for it and coped with the consequences very well.



9 Appendix

9.1 Logbook

9.1.1 Sprint 0

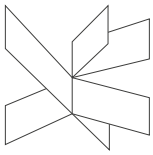
8/8:

We started the project after summer vacation, and DABAI changed its mind about the product of this project, which leads to requirement change:

- As before the system was only intended to be used internally by DABAI for debugging and monitoring purpose. Now the project manager also have vision to sell it to the customers of forecast system. This means the target user has changed from only a machine learning engineer to also include system admins in a hospital. And there will be need to later integrate authentication and authorization mechanism into the system. (This also means, an UXer will be involved in features with GUIs)
- Before the requirement wanted two choices of time interval for monitoring the system: one week and one month. And now the PO wants to change the requirement to being able to monitor the forecast system for historic period with adjustable length, with 2 weeks as default length.
- Before two metrics were required for support in the requirements: MAE and F1. Now PO means the F1 value will not be relevant anymore, therefore only MAE should be supported.
- Before the requirement for displaying model information didn't require displaying the unit name for which the forecast is done, now it is also required to display a readable unit name together with model information.

We re-discussed the hierarchy and terms for the forecast system:

- Forecast system (one per customer)
- Forecast application (one per use case/client)
- Forecast pipeline (one per unit_key)
- One trained prediction model (one model per pipeline and day/specific time interval)

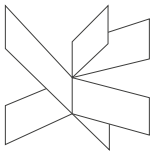


We clarified the most frequent data update rate, which is currently every 10min
We clarified the overall architecture, interface and cooperation between forecast system and our system
We decided to not use any database or no data cashing except client list and unit list
PO promises:

- Interface description for forecast system (API documentation), API should exploit:
    - Forecast system list (customer list)
    - Test data MAE
    - Clients list
    - Unit_keys_list
    - Forecast result
    - Actual result
- Add unit names to the data points based on the unit_key
- Agreement with UX cooperation (UXer join spring review and FDD meetings for GUI features)
- Access to bitbucket, Jira, Team City, Wiki

We decided to have a 2 weeks sprint interval, with Friday as sprint end and start
We choose the library "C3" for data visualization, because it is already used for DABAI
Start sprint 0 which ends 17/8
<u>Goals:</u>

- Set up development environment:
    - Georgi - initialize front end repo and initialize socket communication
    - Ophelia - initialize back end micro service repo and initialize socket communication
    - Tomas - prototyping with Angular and C3 (visualization library)
- Full fill the technology requirements:
    - Every system component in the solution system should be stateless (Ophelia)
    - Front end should be implemented in a stable version of Angular (Tomas, Georgi)



- Data visualization library/framework used should preferable be free (Tomas)

Ophelia is assigned scrum master

We agreed on no specific report writing activity, but using FDD documentation for report chapters

9/8:

Ophelia:

- So far in the sprint I've created an overview of the project and process report sections, I've wrote project initiation and description phase reflection for the process report and rewrote the sections from project description, so it is more accurate and can be used in the introduction section of the project report.

Georgi:

- Today I've received access to the work computer at Systematic and with some help from Ophelia I got familiar with the systems used in Systematic (Wiki, Jira, Bitbucket). After that I've installed the software I need for the development process like Visual Studio Code, Node, Angular CLI, Node version manager.
- After everything was setup I've initialized the front-end repository and generated the base project for the front-end application

Tomas:

- I started prototyping with Angular 2 by creating a "Hello world" application, so I can get familiar with the framework.

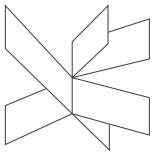
16/8:

Group:

- The team got the initial GUI draft design from the UXer
- The team had a meeting for building the feature list based on the existing requirements from PO together. We defined 8 features and only the first two, that we know we are definitely going to make first are descripted with more than just what requirements we intend to cover.

Georgi:

- I made some initial prototyping for the dashboard page based on the GUI draft that we've received from the UXer.



- After receiving the initial GUI draft I found a missing step. On our first meeting with the PO he requested to have one step for choosing the system (region) and then all the clients should be listed, while on the draft we received that step was missing.

Tomas:

- I've started preparing FDD0000

17/8:

Sprint meeting:

In the sprint meeting we decided that we need a three-level structure for the dashboard:

1. System
2. Application
3. Model (unit)

All of the levels should be in a collapsible sections like a tree structure.

The PO pointed out that logging the activity of the system and testing is implicitly expected from us.

We've received the API documentation for the Forecast API which can be found on: dabai-test1.systematicgroup.local:6060/

We've decided that we are not going to held a kick-off meeting because we already had an initial meeting and the start was discussed

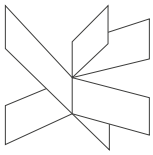
Retrospective meeting:

Georgi:

- One thing I'm not happy about is that we are discussing the same thing over and over again, meeting after meeting and we are not going anywhere with the decision. I felt like the past few meetings the same thing was discussed with the same result and I can't understand why it is still discussing.
- A good thing that I liked during that sprint is how fast we created the feature list, without any problems and it felt like everyone understands what needs to be done.

Tomas:

- I like that we managed to create the feature list rather quickly and it ended up in a good feature list



- I didn't liked the fact the Ophelia decided too many thing on her own without consulting with us, for example he decided to schedule a meeting for feature list building out of the sudden, or after the sprint scope was already defined she asked for other contributions

Ophelia:

- We didn't face any real obstacles until now
- One thing we forgot was to allocate time for administrative work when we planned sprint 0
- I didn't like the fact that the other team members didn't wrote in the logbook

9.1.2 Sprint 1

Planning:

We've decided that Tomas should be the feature lead for feature 0 since the feature is mainly backend related and Georgi should help with both RFC and RFI clarifications because the frontend is concerned as well.

define API first and split design into backend and frontend so Tomas & Georgi can work in team

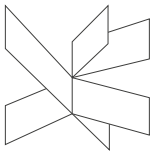
Tasks we added to the sprint backlog:

- Logbook writing - 3h
- RFC for FDD0000 – 8h (Ready for RFC) + 3h (RFC) + 1h = 12h
- Buffer for extra meeting - 1.5h
- RFI for FDD0000 – 8h (Ready for RFI) + 6h (RFI) + 2h = 16h
- Process report - writing group description - 3h
- Project report - analysis for problem formulation - 8h
- Supervisor synchronization meeting - 3h
- Scrum master administration work - 1h
- Prototyping frontend – 8h

22/8:

Ophelia:

- Finished answering most of the analysis questions, except the source system API description, will do that tomorrow.



Georgi:

- Finished with the front-end stories for Sprint 1 with rough estimations
- I'm not entirely sure that the stories are enough or if they are correctly formulated - must consult with the rest of the team
- The rough estimate for the front-end stories ended up being 20, compared to the 50 for the whole feature is good enough because only the first story can be implemented in parallel with the backend since the rest requires the web service to be functioning in order to retrieve the data

Tomas:

- Backend stories for FDD0000

24/8:

Ophelia:

- Added more detailed description of forecast API to the analysis part.

Georgi:

- I've created a list with the feature stories in the order they have to be implemented

12.1. Story 1: Dummy REST Web Service
12.8. Story 8: Implement a service to retrieve data from the web service
12.9. Story 9: Display the data retrieved from the web service on the web app
12.2. Story 2: API Help Page
12.3. Story 3: Installations endpoint returns live data
12.4. Story 4: Clients endpoind returns live data
12.5. Story 5: Units endpoint returns live data
12.6. Story 6: FS results caching and job scheduling
12.7. Story 7: Building web app's layout based on the GUI draft (with dummy data)

Tomas:

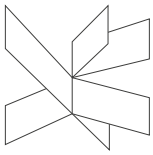
- Sick

29/8:

Tomas:

- I've finished with the backend part of the RFI for FDD0000
- I've updated the logbook for Sprint 1

Georgi:



- After the daily scrum we have decided that we need one more story for creating a translation service for the frontend in order to fetch text values from a translation file instead of hard coding them.

Ophelia:

- Discussed the argument for architecture choice with Tomas
- Finished group description.

31/8:

Group:

We've held an RFI meeting for FDD0000

- The RFI meeting was helpful to clarify everything about the first feature and also make somewhat more accurate estimations
- The Sprint retrospective went well we presented our first feature with the feature stories to Alan and he was happy with them. He pointed out some small things we need to be careful about but there were no issues whatsoever with the stories.
- Now we are ready to start the implementations with the next sprint

Sprint retrospective:

Tomas:

- I think that everything went pretty well and we reached our goal to get FDD0000 RFI-ed
- Estimation for the tasks could be better

Georgi:

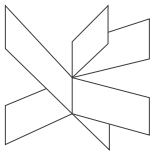
- I think that we underestimated our capabilities for how much we can do in this sprint which caused a low productivity for this sprint.

Ophelia:

- Only wrote report this sprint, it was bit boring, otherwise it's fine sprint
- I would like to be involved more like a tester in the future rather than writing report

9.1.3 Sprint 2

4/9:



Ophelia:

- I've prepared FDD0001 for RFC

5/9:

Daily Scrum:

- In the daily scrum Georgi and Tomas discussed if the GUI design should strictly follow the drafts from the UXer

6/9: (Initial supervisor meeting)

Outline:

Part 1: How do we work together

How often do we meet:

-> Every sprint Demo and retrospective (bi weekly on Fridays at 1pm)

In person or Skype:

-> Via Skype, change it if won't work out

What should we focus on:

-> Analysis, Design & Test + reasoning for our choices and considered alternatives

Exam questions:

-> Usually derived from project report

Part 2: Reports

Are all parts from the template relevant

Don't make documentation just for the sake of documentation, FDD is fine (i.e., no need for Use case diagrams if we don't use them)

In process report, Reflection, Group description and discussion do make sense, for the logging part, logging sprints rather than days is enough

Part 3: Ole's comments:

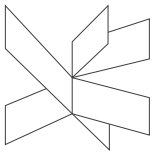
Seems like a good project to do

For the analysis section you should move the technical API description to appendix and only mention the usage of REST. Also, the technology used is a part of design and not analysis.

14/9:

Ophelia:

- I worked on RFC for FDD0001



    - Suggested scope change
    - GUI indication
- Check on testing for front-end stories for FDD0000

Review:

- GUI demo:
    - Unit_name not found scenario:
        - For the scenario where the unit name is not found the front-end should show "Unit name not found" plus the unit ID so it can be identified in the database
    - Tooltips
        - The tooltips design should be changed so it matches the CSL theme
    - Hierarchy with region
        - The front-end should follow the tree structure for listing Installations -> Clients -> Units

Retrospective:

Nothing much to say about the sprint except that we need to make better estimation on the stories and maybe change to hours instead of story points

9.1.4 Sprint 3

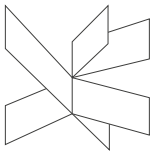
Planning:

More focus on getting FRI so we can always code

We had a discussion about the socket connection and how should we implemented because we had it planned for Sprint 0 but it was not implemented then but it is part of FDD0000 so we had to plan and implement it before we can close the feature

Ophelia:

- Ready FDD0001 for RFI
    - I've created a Sequence diagram and wrote the Technical design section
- Test FDD0000
    - I've design and performed some manual tests



- FDD0001 requirement change, requires rework on sequence diagram and the technical design
    - Change on calculation logic for performance indicator

Review:

During the review Georgi showed a demo of the socket connection between front-end and back-end and also the use of translation service on the front-end.

Some of the tasks planned for this sprint wasn't done so we discussed the reasons:

- Feature test task was not finished because there was no test environment providing fixed test data
- Feature close task was not finished because Tomas went on a vacation
- Ready FDD0001 for RFI – During the RFI meeting we found out that we need to change some of the requirements
- Ready FDD0002 for RFC – There was a bit of confusion about what needs to be in scope for that feature but in overall the feature is ready for RFC meeting

Retrospective:

Lack of knowledge sharing for backend implementation is causing bottle neck. We need better knowledge sharing strategy in the future. Supervisor suggests more diagram to ensure common understanding of the architecture.

As a result, we added task for documenting current architecture in the backlog

Ophelia feels it is difficult to manage both being feature lead to start up the next feature and being the tester who can close the previous feature, which is also becoming bottleneck for the team.

We forgot to think about a generic test strategy for the project, this must be done ASAP.

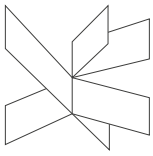
9.1.5 Sprint 4

Planning:

PO priority:

- Closing current feature

The current feature list and requirements are no longer fully correct, we need to review them



- As a result, we added task for this to the new sprint

Documentation and fine writing FDD sections for report purpose is prioritized, as it is recommended by the supervisor

Initial test strategy meeting should be held by Ophelia

Set up of test environment is prioritized as it blocks the feature close for FDD0000

Our logbook requires a lot of implicit knowledge to understand, the supervisor would like it to be easier to understand for him, so it is also prioritized to "fix" the previous writings there.

Ophelia:

- I have reworked the sequence diagram for the performance indicator calculation logic for FDD0001
- I was hospitalized

5/10: (Feature roadmap and requirements review)

Ophelia was absent because of sickness

Reprioritized the feature list

Inserted new feature FDD0006 Display client prediction performance based on the UX-design

Added new requirement PP10 to the requirement base

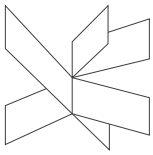
The solution system can display prediction performance for all units within a client using heatmap

12/10:

<u>Review:</u>

Project Report comments from supervisor:

- Analysis:
    - Write a section about what kind of product we want to build
    - Make use case diagram based on the feature list
    - FDD documents as appendices
    - FDD process (incl. template etc.), into Process report
    - Finish answering questions from analysis
- Introduction:
    - Table Choice of models and methods as appendix
- Design:



    - Pattern diagrams
- Implementation:
    - Describe different used packages (why, what for, etc.)
- Test
    - Describe the test process from FDD

Retrospective:

Ophelia's health condition became a liability for the project, but the team and supervisor had agreed to give her one more chance

She'll no longer be a feature lead or participate in implementation or be a scrum master and spend more time with documentation and testing

Tomas takes over Ophelia as the new scrum master

9.1.6 Sprint 5

Ophelia:

- Update diagram for FDD0001 again, now split the background job into two jobs, one for updating the data cache in general and one for model performance calculation logic

Georgi:

- Finish RFC for FDD0002
- Prepared the front-end stories for FDD0002 RFI

26/10:

Review:

Comments from Supervisor:

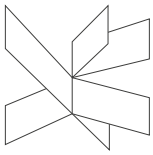
Extend REST API analysis:

- Add relational diagrams
- Add some Use Case diagrams to make it more clear what are we building (dashboard)

Other commets

- Otherwise good progress
- Expect to hand in 60-90 pages (currently at 30)

Retrospective:



We should have daily scrum every Friday so the progress can get tracked.

Focus on non-blocking FDD stories

Ophelia can increase workload by 0.5 day/week

9.1.7 Sprint 6

Requirement change:

The product owner requested change in the requirement PP4. So, the feature FDD0001 has to change implementation logic before closing, which increased the scope of the sprint.

The old version of the requirement sounds:

The solution system should be able to display performance information in a sorted manner, so the presentation of Region, Client and Unit are ordered by MAE in descending order.

The new version of the requirement changes the sorting key:

The solution system should be able to display performance information in a sorted manner, so the presentation of Region, Client and Unit are ordered by first the severity of the performance indication, then MAE, both in descending order.

Group:

Design section review meeting:

- The authors of the current content presented what they wrote
- Talk about what more should be in the design section
- Talk about what should be moved to other section of the report

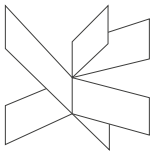
Ophelia:

- Implements the changes to Analysis section requested by the supervisor
    - Use case diagram
    - Source data structure diagram
- Design test for FDD0002
- Help closing FDD0001

Review:

Ole wants access to our project report

We agreed on FDD004 as last feature to be implemented



9.1.8 Sprint 7

Goal:

Close FDD0001

Finish Project report - Analysis & Test

Group:

- Sprint scope increased during the sprint

Ophelia:

- Report analysis section - make use case description for all of the stated use cases
- Report test section - add code examples and presentation of the test design table

Product owner meeting:

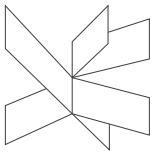
Review of FDD0002 historical data vs. predictions visualization

- Discussed the scale of the chart
- The PO accepts the result

PO generally satisfied with our progress and that FDD0004 is the only new feature to be added in this sprint.

Supervisor meeting:

- Analysis:
    - Use case diagram too complicated
        - Simplify the diagram, move the complexity to the use case description
    - Forecast system data schema
        - Describe each entity
        - Explain the difference between key and key_id
        - Explain the relationship type, especially between the link of client and unit, why it is not an aggregation
    - Technology
        - Reference the design section for the choice of Angular version
- Design
    - More diagram for overall system architecture



    - Introduction for the design section
    - Example for how SOLID is implemented in our design
    - Patterns -> why do we choose to use them?
- Test
    - Use "validate" instead of "Prove"
    - Elaborate about early testing in the process report
- Process report
    - The most important things:
        - Which process did we use:
            - FDD
            - Scrum
            - Early testing
        - Log book
        - Personal reflection

9.1.9 Sprint 8

Goal:

Close FDD0002

Finish reports

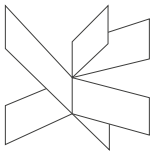
Ophelia:

- Correct comment for analysis and test section
- Write project execution section for process report
- Write own personal reflection for process report
- Write supervision section for process report

29/11: (Project Report – design feedback)

Backend:

- Data Access layer:
    - Start with what we do
    - Data consistency – rephrase
    - DAO - explain its just a prefix, not a DAO pattern



- Domain logic layer
    - Move FS to DAL and simplify diagram

Frontend:

- Use academic writing style (no pronouns)
- UI explanation

4/12: (Process and Project report feedback)

Project Report:

- MVVM, rephrase into something like model-view separation, angular 2 is not MVVM
- Reactive programming part: more on WHY (Honestly I'm not so sure)???
- More information what Reactive programming is and reference to the Implementation section

Process Report:

- Remove specifics about the illness
- Perhaps better to swap sick for ill
- Enhance FDD with how we used it (maybe share some notes, jira board, or burndown chart)