Project Plan

Mobilefant

December 9, 2013

Contents

1 Introduction

Agilefant is an open source tool for agile project management. Currently it is provided as an open-source version and a hosted version. The hosted version comprises more features in comparison to the open-source version.

Agilefant has approximately 10,000 users worldwide, and according to the customer, the number of registered users increases every day.

Agilefant is a very powerful tool for requirement management but currently it is too detailed to be used on mobile devices (small screens). The customer wishes that the users of Agilefant could use its the most important functions using their mobile phones and tablets. For example, users could log spent hours after a workday when sitting in a bus. Agilefant's main competitors are already providing mobile applications, so it is crucial to Agilefant to also provide a mobile client. Therefore, the goal of our group is to develop a mobile application that works along the hosted version of Agilefant and can be used on both smart phones and tablets.

1.1 Vision

Customer's vision is to become the leading provider of agile backlog management tools. This mobile application has an important role of fulfilling the vision as missing mobile support can be a major threshold factor for many potential new users.

2 Stakeholders and staffing

The project contains several stakeholders, which are presented in Figure 1. The stakeholders are divided into four main groups: the customer (Agilefant), the student group, the teaching personnel in Aalto University, and teaching personnel in University of Victoria (UVic). In addition to these, Amazon's web server is listed as a stakeholder as the application needs to communicate with it. Arrows in the figure present the direction of main communication.
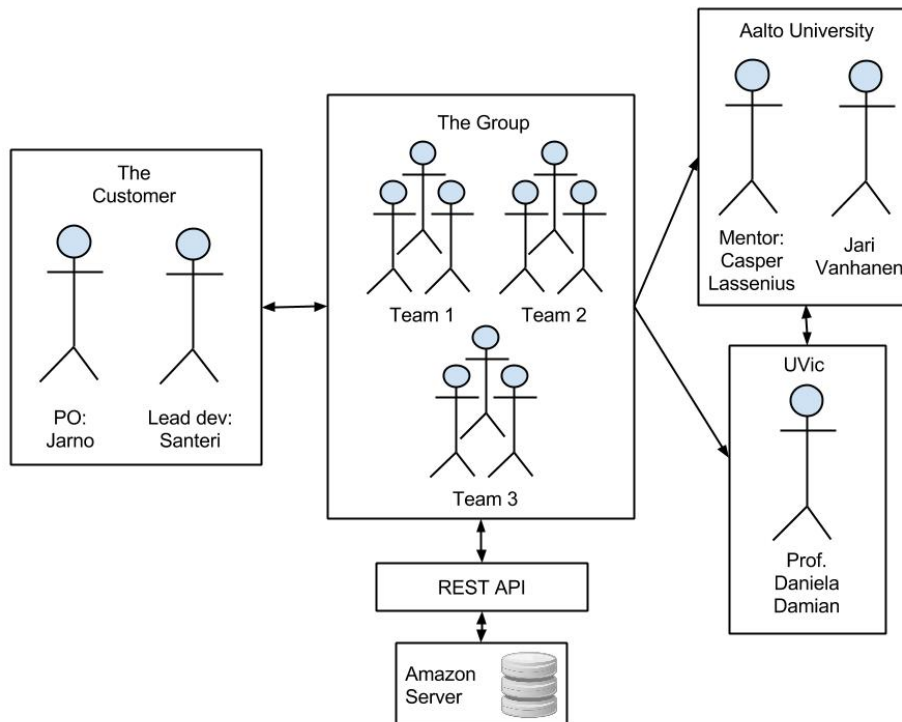
Figure 1: Stakeholders of the project

2.1 The Team

At the beginning of the course we had a team of 9 people, but we lost one developer so now we have only 8 students. At the beginning of 2014, 15 Canadian students will join us, so we will have a group of 23 students. We will form three teams, each of them consisting of both Finns and Canadians.

In this document, only some personal information is published. We have a document with stakeholders' more detailed personal information but we don't want to make that public.

The group's webpage is https://github.com/phyper/mobilefant-documentation/wiki and its email is mobilefant#mobilefant.flowdock.com.

We have several roles and one person can have several roles. The primary role given by the course is bolded. The roles are:

- Project Manager = PM

- Scrum Master = SM

- Lead Architect = AR

- Quality Assurance = QA

- Requirements Engineering = RE

- Developer = Dev

Role	Name	Email	Responsibilities	Assistant role
PM, SM, RE	Benjamin Behm	benjamin.behm#aalto.fi	Organizing the work, removing impediments, documentation, process supervising, eliciting requirements, working as a scrum master	-
AR, RE	Harri Lampi	harri.lampi#aalto.fi	Architectural design, eliciting requirements, coding, UI design, documentation	-
QA, RE	Matias Kuusela	matias.kuusela#aalto.fi	Quality assurance, eliciting requirements, UI design, documentation	-
Dev	Miro Vilkki	miro.vilkki#aalto.fi	End-to-end development	AR
Dev	Rolle Saarinen	rolle.saarinen#aalto.fi	End-to-end development	AR
Dev	Janne Gröndahl	janne.grondahl#aalto.fi	End-to-end development	QA
Dev	Janne Kajovuori	janne.kajovuori#aalto.fi	End-to-end development, documentation, UI design	PM
Dev	Joakim Kronqvist	joakim.kronqvist#aalto.fi	End-to-end development	AR

Table 1: The Finns

2.2 Mentor

Role	Name	Email
Mentor	Casper Lassenius	casper.lassenius#aalto.fi

Table 2: Mentor in Finland

2.3 Customer

Role	Name	Email
Product owner	Jarno Vähäniitty	jarno#agilefant.org
Tech. Lead	Santeri Korri	santeri#agilefant.org

Table 3: Customer representatives

2.4 Teaching personnel in UVic

Role	Name	Email
Professor	Daniela Damian	damian.daniela#gmail.com

Table 4: Staff in University of Victoria

3 The Goals

3.1 Project goals

The main goal is to develop a reliable mobile application for Agilefant that contains the main functionalities of its cloud version and fulfil customer's vision of the product. Furthermore, the goal is that everyone's personal goals will be reached and the course has been an educational experience. Other high level goals are identified, and these are presented in Table 5.

#	Goal	Verification Criteria
1	To produce high customer satisfaction	Customer's personal opinion about the delivered product.
2	To implement a limited set of key use cases	Architecturally sound, clear implementation and testable.
3	The product will be released after the project	Whether the customer release the product or not.
4	To produce an application that is easy to develop further	The customer's developers understand the code easily and they have no difficulties to add new features in it.
5	To get grade 5	The grade will be visible in transcript of records or the course personnel has verified the grade.
6	To win the quality award	Our group has selected as the best group at the end of the course.

Table 5: Project goals in a priority order

3.2 Personal learning goals

As we are here learning new things, we should focus on to learn things we are interested in. Thus, it is important that everyone tells theirs interests aloud and points out what they would like to learn during this course. In this reason, everyone has written down their personal learning goals. To reach these goals, people should try to participate in tasks that support their learning goals. If a developer is going to take this course second time later and has a preferable role in his mind, it is really recommendable that he helps that SE trio's member.

4 Resources

4.1 Personnel

The course in Finland is a course of fixed hours, meaning that the Finns must invest "credits * 27 hours - 15 hours" in the project. For example a person with 8 credits must spend 201 hours during the project.

In Figure 2, we have estimated hours everyone is going to use during the course. The same information is available in a time allocation spreadsheet. In that spreadsheet we also follow how much effort has been used and how much is left.

Member	ECTS	Hours		Effort Left	Project Planning 46[1]	47	48	49	50	PP Total	Christmas Vacation 51[2]	52	1	C Total	Implementation 1 2[3]	3	4	5	6	7	8[4]	Imp 1 Total	Implementation 2 9[5]	10	11	12	13	14	15[6]	Imp 2 Total	Grand Total
Benjamin Behm	8	201	Planned	0	15	15	10	10	10	60	0	0	7	7	30	30	7	7	7	7	8	96	5	5	5	5	5	5	8	38	201
			Realized	143,5		24	10,4	11,4	11,7	57,5				0								0								0	57,5
Matias Kuusela	8	201	Planned	0	8	8	10	10	10	46	0	0	7	7	15	15	10	10	10	16	12	88	8	8	8	8	8	8	12	60	201
			Realized	164,75	5,5	5,75	19,3	5,7		36,25				0								0								0	36,25
Harri Lampi	6	147	Planned	0	6	8	10	10	6	40	0	0	7	7	8	8	6	6	6	8	8	50	8	8	6	6	6	8	8	50	147
			Realized	105,5	1,7	13,1	19,2	7,5		41,5				0								0								0	41,5
Miro Vilkki	5	120	Planned	-8	4	4	4	2	2	16	0	0	0	0	8	8	8	8	8	8	8	56	8	8	8	8	8	8	8	56	128
			Realized	105,3	1,7	3,5	4,5	5		14,7				0								0								0	14,7
Rolle Saarinen	5	120	Planned	0	4	4	4	2	2	16	0	0	0	0	0	9	8	8	8	8	8	49	8	8	8	8	8	8	7	55	120
			Realized	109,8	1,7	5,5	3			10,2				0								0								0	10,2
Janne Gröndahl	5	120	Planned	-8	4	4	4	2	2	16	0	0	0	0	8	8	8	8	8	8	8	56	8	8	8	8	8	8	8	56	128
			Realized	103,3	1,7	3,5	4,5	7		16,7				0								0								0	16,7
Janne Kajovuori	5	120	Planned	0	4	4	4	3	3	18	0	0	2	2	8	8	8	2	8	8	8	50	8	8	8	8	8	8	2	50	120
			Realized	87,15	3,8	9,5	11,3	8,25		32,85				0								0								0	32,85
Joakim Kronqvist	6	147	Planned	0	7	6	6	6	6	31	0	0	4	4	8	8	8	8	8	8	8	56	8	8	8	8	8	8	8	56	147
			Realized	117,55	7,6	10,1	4,5	7,25		29,45				0								0								0	29,45
Total			Planned		52	53	52	45	41	243	0	0	27	27	85	94	63	57	63	71	68	501	61	61	59	59	59	61	61	421	1192
			Realized		47,7	61,3	77,7	52,4	0	239,15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	239,15

Figure 2: Time allocation (updated on 2013/12/06)

In a Figure 3, it can be seen that total hours used in the planning phase is almost equivalent with the planned hours. The figure doesn't count hours used in the last week of the planning phase.
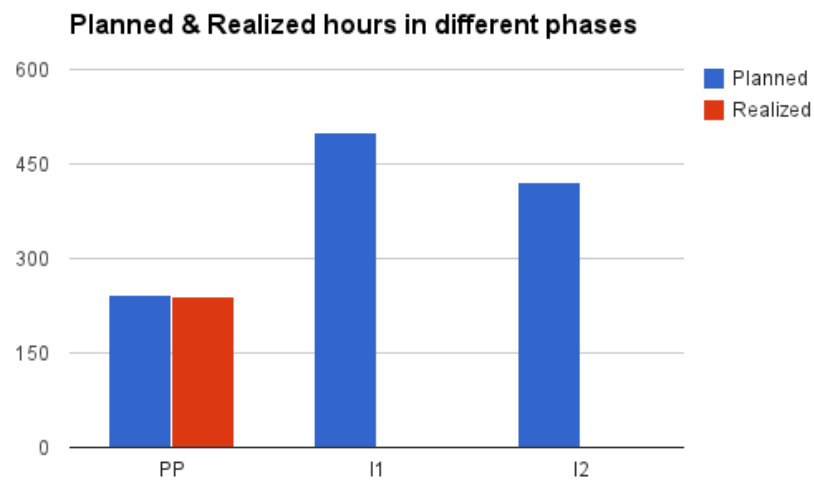
Figure 3: Hours per a phase (updated on 2013/12/06)

Figure 4 shows the planned and realized hours a weekly basis. It is clearly visible that after the first week the realized hours exceeded the planned hours. The Finns should monitor used hours to be sure that hours won't run out in the middle of the course. The project manager monitors regularly the spent effort.
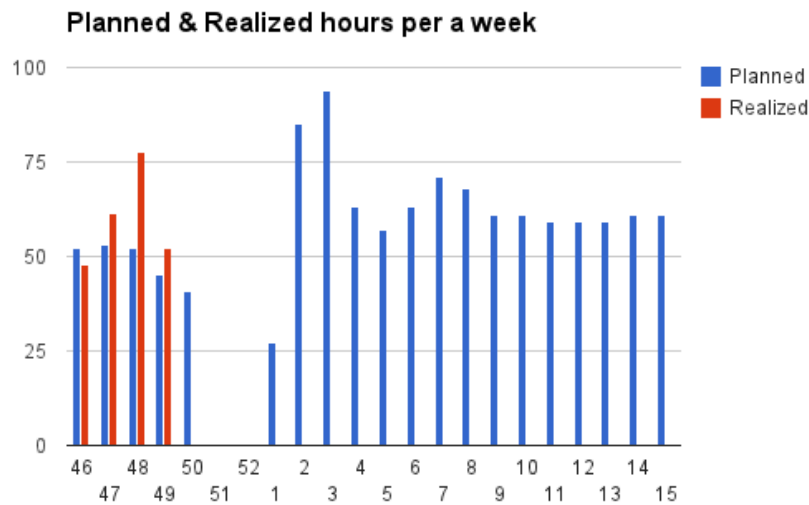
Figure 4: Hours per a week

Hour estimates will be adjusted again at the beginning of Sprint 3 when students know their spring's schedule and we get more information about Canada cooperation.

4.2 Material

Development tools

Development tools used during the course are IntelliJ IDEA 12 Ultimate, Apache Cordova, Android SDK, XCode (on Mac), and Github. The project manager has a JetBrain's Classroom License so IntelliJ's full version can be used.

Communication tools

Flowdock is used as a primary communication tool. Aalto provides 180 days license for that. Google Hangout will be used as a video conferencing tool.

Design tools

Lucidchart tool is used to draw UI wireframes and other diagrams.

Working space

Development will take place at CS building in the room A243. The room is shared with an another project group (#15 - TrafficSense) so we have been scheduled the usage of the room with them. The idea will be that the both teams will have specific days and hours the room is exclusively reserved for them. At other times, everyone

could use the room. Currently we have agreed that our team uses the room on Monday and Thursday, and the another team on Friday and Tuesday or Wednesday. We have shared our calendar with them so they know when the room is available.

The CSE department borrowed three desktop computers to our group with Ubuntu 12.04 installed. These computers have been set up to our team room.

Testing devices

We need mobile devices to test the application. The customer has promised to deliver some test devices, but the selection will be limited. In this reason we will also use our own mobile phones for testing. We have phones with IOS, Android and WP8 platforms.

Agilefant's REST API

The application must communicate with Agilefant's back-end. Agilefant contains a REST API that we use to manipulate data on Agilefant's server. The REST API is not totally ready yet but the customer has promised to put some effort to make it work, if needed.

5 Work practices

In this section we describe what working practices we have planned to use in this course. Each group member should understand what practices are used and how to adopt them properly in order to work efficiently.

5.1 Practices

5.1.1 Iterative development

We are working iteratively during the course. The idea is to build the product incrementally and iteratively. We will be using an agile software development method called scrum that is presented in Figure 5. We will be following the scrum practices meaning that we will keep sprint planning, sprint review and sprint retrospective sessions in every sprint.

Even though the course is divided into three phases (Planning, Implementation 1 and Implementation 2), we can't rigorously follow these phases because of the Canada cooperation. We have planned to have two sprints before the Christmas and six or seven sprints after the Christmas. As we are not aware of how the first sprint with the Canadians will be held, we cannot plan next year's sprints yet. The last sprint will be devoted for bug fixing and releasing the app.

Because of Finns' exam week on week 8, we might have one sprint of three weeks whereas the other sprints are two weeks.

In general, the Finns will have a common development days on Monday and Thursday. This is because Canadians have their lecture hours on the same days.
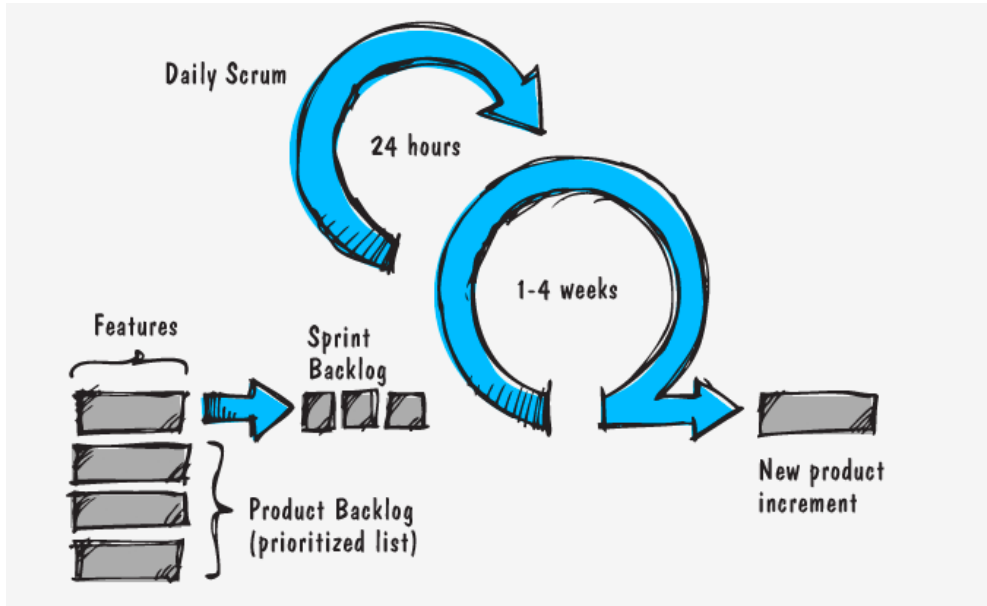
Figure 5: Scrum process

5.1.2 Distributed development

As mentioned earlier, teams will be formed again on January when Canadian students start their course. After that, three globally distributed teams will be formatted as show in Figure 6. This means that we will have three scrum teams with 7 or 8 persons in each. Team formation session will be held in Canada on January and the Finns will participate in that session using video conferencing system (Google Hangout). The goal is to build three equally strong scrum teams which all are able to do end-to-end development.

Teams will be using internet chat, video conferencing and other communication tools to work with other students who are not in the same site. We engourange students to try out distributed pair programming so that there would be Canadian-Finn pairs, but of course this is challenging because of major time difference.
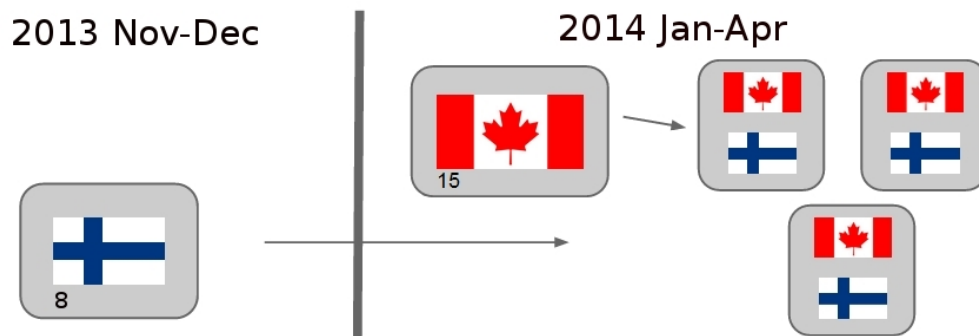
Figure 6: Canada cooperation

5.1.3 Sprint grooming

A couple days before the end of a sprint, the project manager grooms the project backlog with the product owner. Also other team members are allowed to participate in the meeting if they have time. In this meeting they will prioritize the project backlog and clarify user stories so that the project backlog is ready for the sprint planning meeting.

We have agreed with the customer that it's up to product owner to ensure the project backlog is up to date. Teams have no permission to change the priority of stories in the project backlog. In addition, the product owner has no permission to push stories to sprint backlogs, as it's developers' tasks.

5.1.4 Sprint planning

Sprint planning session will be divided into two parts. The content of the sprint planning is presented in Table 6. The project manager is responsible for arranging sprint planning sessions.

The sprint planning session will be held using Google Hangout so that The Finns are located in their team room in Finland and the Canadians are located in their class room in Vancouver, Canada. In the phase 1, everyone is participated in the same Hangout, but in the phase 2, teams discuss using separate Hangouts.

Part	Duration	Description	Participants
1	1h	The product owner presents the prioritized project backlog, so that the teams would understand what should be done during a sprint. The product owner is there for answering any questions the teams would like to ask relating to the user stories and tasks. In this part top most user stories from the project backlog are estimated using planning poker method. We don't have cards but we use hands and fingers to show the estimates. The product owner is leading this part. Sprint goal is also agreed in this part. When enough stories are estimated, it's time to start the next phase.	Product owner, members of each team
2	1h	After the first phase, teams separate to plan what they can do during the sprint. The scrum master is responsible for faciliting this phase. The product owner should be available for teams if they need to ask something. Teams start with pulling stories from the product backlog to the sprint backlog based on their knowledge of how much work they are capable of doing during a sprint. When there are enough stories in the sprint backlog, it's time to plan how the chosen work will be done during the sprint. Users stories will be assigned to team members. User stories are split into tasks and the required time per a task is estimated by a person the task is assigned to. In this meeting, the team can already start design the system so that they are able to convert the backlog items into a working software increment.	Team members

Table 6: The content of a sprint planning

In a sprint planning session (excluding sprint 1), stories are estimated using story points following the fibonacci scale. Possible story points are 1, 2, 3, 5, and 10. If the story is estimated to be larger than 10 story points, it can be seen as an epic and should be split to smaller stories so that it can be finished during the sprint.

As we have only one product owner and we will have 3 distributed teams, the planning poker could be challenging to be performed using only hands. In this reason,

we have created a planning poker spreadsheet that can be used for several teams. The idea is that everyone opens the spreadsheet and type the number under his/her name, but not press the enter yet. As all estimates has been entered (each cell is gray), the product owner will count 3, 2, 1 and then everyone press the enter and the story points are revealed.

The customer said we should not have larger stories than 10 points. We also agreed that we won't tie story points to any hour estimates. If one story is estimated to have one story point and an another story is thought to be about twice as laborious as the first one, it will be given two story points and so on.

5.1.5 Sprint review

Sprint review is held at the end of each sprint. All teams participate in this meeting. Google Hangout is used to reach everyone. The goal of a sprint review is to demonstrate to all stakaholders what the teams have accomplished during the sprint. The meeting should be informal and PowerPoint presentations are NOT allowed. Teams usually presentes new functionalities they have implemented and these are discussed with others.

5.1.6 Documenting

All course documents need to be public and available for everyone. The course personnel and customer should be able to follow the progress of the project by seeing our documentation. In addition, the change log of each documents documentation should be visible so that changes are easily seen. The official documentation is written in English.

After a long discussion we deciced that we use Google Docs for course documentation except the project plan that was started to write using Latex. All project related documents are stored in Google Drive, to where all stakeholders have access.

The SE trio is responsible for writing the course documentation. Developers can help to write documents based on their interests. Project manager is responsible for writing the project plan and progress report; the QA is responsible for writing the QA plan, requirements document, and testing material; and architect is responsible for documenting the architecture, writing technical specifications, and user's manual.

5.1.7 Risk management

Trio held a risk management session together with one developer. The session was held in the middle of the first sprint and it took 30 minutes. During the session several potential risks were identified. When the risks were identified, we estimated the probability and severity of each risk and discussed a little bit how to minimize the realization of the risks.

The risk log is maintained regularly by the SE trio. At the end of each sprint the log is review and updated.

5.1.8 Time tracking

The group's time tracking will be applied in Agilefant. The group should follow these time tracking practices:

- Each group member should enter their own hours by themselves in Agilefant.

- Hours are logged directly to the story or more preferably to the task after the work is done.

- Hours should be logged before leaving the office

Agilefant provides burndown charts that are used to follow the project's progress. Burndown charts tell also whether estimated hours are correlating with actual hours. That helps the group to shape its task estimation.

Agilefant cannot provide all relevant metrics so the project data is exported to Excel every Monday. The project manager follows that time is logged in Agilefant and remind other team members to log hours if needed.

As the group is logging spent effort to the Agilefant, the customer is able to follow that the project is progressing.

Below are two screenshots of how to log hours to Agilefant. Hours can be entered by clicking hours number on Spent column or by clicking Spent effort under a Edit button.

Figure 7: Story and task with spent effort

Figure 8: Log spent effort

When the course is over, the Finns will be get credits based on the hours logged to the Agilefant (+ hours spent on lectures). The view shown in Figure 9 can be found in Agilefant under Timesheets where user needs to select backlog(s), interval and user(s) to generate the timesheet where used hours are listed.

Agilefant	226h 27min
SuperAmerica	**129h 47min**
Direct spent effort	1h 40min
Child backlogs sum	128h 7min
Training sprint for Agilegret	**1h**
Direct spent effort	1h
Agilegret - Sprint 1	**16h 6min**
[+] Stories	5h 1min
[+] Tasks	4h 40min
Direct spent effort	6h 25min
Agilegret - Sprint 3	**37h 30min**
[+] Stories	29h 31min
[+] Tasks	4h 54min
Direct spent effort	3h 5min
Agilegret - Sprint 4	**25h 45min**

Figure 9: Total used hours

5.1.9 Communication

Teams will keep a daily scrum meeting during each development day. Before the Canadians join us, daily scrums will be held in our teamroom. In spring, daily scrum meetings are held using Google Hangout so that the whole team participates in them. The daily scrum will be a short, 15-minute time-boxed meeting where team members synchronize their activities. In this meeting, people will tell, in turn, three things: What they have done since last daily scrum, what they will do before the next meeting, and whether there is any obstacles in their way. Attending to these meeting is important in order to know what is happening within the team.

Scrum of scrums will be held when three teams have been formed. These meetings are arranged once or twice per a sprint. The idea is that one member from each team will keep a short meeting together and share information what each team has been doing. This helps to solve possible roadblocks that can emerge when several teams are developing the same application.

We can also try out a practice where a team member attends to other team's daily scrum to get information about what they are doing. Then the person can share the information with his/her team.

Flowdock is agreed to be used as a primary communication channel. It allows to create own communication flows for each team. In addition to team flows, technical discussions can be separated into own flows, so that all parties who are interested in the subject can participate in the conversation. These can be seen as communities of practice. Currently we have separate flows for documentation, UI and REST.

Google Hangout is proposed to be used for video communication with off-site team members.

Project manager send a weekly email to all stakeholders to keep them informed

about the progress of the project.

In very urgent situations phone calls or text messaging can be used, but primary the group is using tools mentioned above.

A public Google calendar is used to share all events and deadlines of the project. Each team member and other stakeholders are able to see the calendar and follow our meeting times.

Communication with the customer usually happens face-to-face in our teamroom or their office.

5.1.10 Defect tracking

For defect tracking we are using Github's Issues application. It is an easy-to-use tool for keeping track of issues found during the development process. It allows to assign a person to the issue and define a milestone when the issue needs to be fixed. It also informs us immediately when new issues are created. This is one reason why the team wanted to use some real issue tracker instead of Agilefant. Issues are also automatically posted on our Flowdock stream from where everyone is able see them.

When a new bug is found, it need to be reported immediately to the issue tracking system in order to inform everyone else of found issue. The issue can be closed only when it is fixed and someone else has tested it.

5.1.11 Version control

All code should be located in the version control system. Agilefant uses Git (and Github) so we are also going to use them. Soberit provided us a private repository. The repository is named as mobilefant.

We have decided to try out feature branching approach with a single repository. In feature branching each feature is developed using a separate branch in the version control system. This means that every time a developer starts a new feature, he/she will create a new branch for that, and after the feature is ready, the feature branch will be merged to the main branch. The main branch should contain only working code, which is verified by compiling the code and running the test. All tests should pass.

We have to be sure everyone understands how to use Git in order not to mess the whole repository as everyone has privileges to push anything into master. To help people to work with Git, we have defined a general git workflow.

To keep the commit message history clear, everyone shouls use the imperative present tense in commit messages. Furthermore, the commit message should start with a short summary of changes that can be up to 50 charts long. More information about development practices can be found here.

5.1.12 Process improvement

A retrospective meeting is arranged at the end of each sprint. The goal of having regular retrospectives is to improve the process and find out the problems before they cause any bigger problems.

For retrospectives, we are using ARCA-tool developed in Aalto University. It's a tool that helps us to keep better retrospective meetings and analyze problems more deeply in comparison to the retrospectives kept using Post-it notes or Spreadsheet. In this way, these meetings can be more productive, worthwhile and less time will be wasted.

The retrospective meeting is time-boxed to half an hour. We have to have enough time to sit down and discuss about the past sprint, otherwise there is no reason to keep this kind of meeting.

The retrospective contains five phases:

1. First, everyone enters problems to ARCA-tool. (5 min)

2. Second, the entered problems are discussed together. (5 min)

3. Third, everyone enters underlying causes to the problems identified earlier. (5 min)

4. Fourth, causes are discussed and additional causes are added if needed. (5 min)

5. Fifth, findings are summarized at the end of the meeting. (10 min)

Figure 10 and Figure 11 show how to use the ARCA tool and how the data can be analysed.
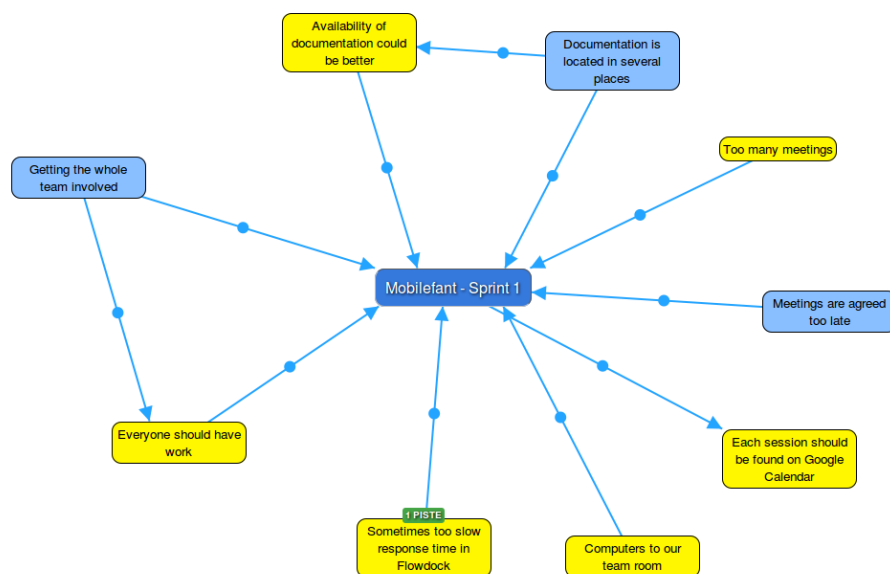
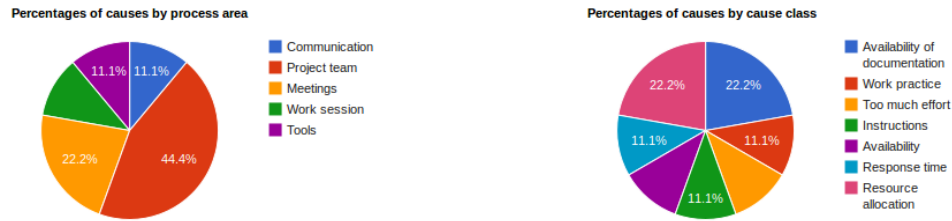

Figure 10: Retrospective - sprint 1

Figure 11: Retrospective analysis - sprint 1

5.1.13 Requirement engineering

Requirement engineering is up to the SE trio and the produt owner, but other team members can also participate in the eliciting process. At the beginning of the project, a requirement engineering session was held with the customer where high level requirements were defined. In general, the product owner has the vision of what should be developed and it is up to him to update the project backlog. The team generates ideas at the same time they are implementing existing user stories. The ideas are shared with the customer face-to-face or in Flowdock.

As we are working an agile way, we don't define requirements at a detailed level at the beginning of the project. But instead, requirements emerge as the project elapses. The most important requirements are specified and prioritized in a backlog grooming session at the end of each sprint so that the project backlog would be up to date in the sprint planning session. All requirements are collected in Agilefant and it is up to the product owner to keep the project backlog up to date and prioritized.

Requirements are usually listed in the project backlog in non-user story form to keep the list clear and readable, but the team together with the product owner should define the user stories in the description field for each story. This format helps everyone to capture the who, what and why of a requirement. The template of a user story is presented below:

```
As a <role>, I want <goal/desire> so that <benefit>.
```

In a user story, the role and goal/desire are mandatory, but the benefit part is optional.

5.1.14 Design

Overall architectural design is up to the lead architect, but other teammates are also allowed to participate in the design process. The customer's lead developer is actively participating in the architectural planning as he has the technical view of requirements. As we need not to touch the Agilefant's backend, we need not to have a detailed view of what is happening in there. We need to design what technologies are used so that the performance would be good enough and it follows the techological practices used in the Agilefant's hosted version.

UI design sessions are arranged at the beginning of the project. First, the UI wireframes are drafted using pencil and paper, but after the drafts are approved by the customer, they are transferred to digital form using Lucidchart tool.

Two team members were chosen to take the responsible for the UI drafts as they pointed out they would be willing to do those. More UI wireframes are drawn as they are needed.

5.1.15 Coding standards

To produce clean and understandable code, we use coding standards defined by Google. Style guides for HTML, CSS and JavaScipt are the most important ones everyone should be familiar with.

5.2 Quality assurance plan

Quality assurance plan will be written later by QA.

6 Phasing

Tasks are not listed in this project plan, as they are listed and maintained in Agilefant.

As presented in Scrum, a product owner is responsible for keeping up the project backlog. The students pull stories from the project backlog to sprint backlogs which are listed below:

- Sprint 1
- Sprint 2

6.1 Schedule

The spring's schedule is still an open question as we don't know how we are going to organize the third sprint in Canada and when the fourth sprint should start. We need to discuss about this with our mentor and the Canadian professor. There are three optional schedules that are presented below. In the first one we would hold the sprint planning sessions on Mondays, while in the others it is on Thursdays. In the second one, we have only sprints of two weeks after the Christmas, excluding the sprint 3. The last one takes into account that we have an exam week here in Finland on week 8, so the sprint 6 would be 3 weeks long.

Project manager and QA are flying to Canada with the customer and mentor on 8.1.2014 - 20.1.2014 (preliminary days). Their task is to help Canadian students to get in to the project by teaching them what Agilefant is and what processes we are using. This is kind of kickoff session for them.

Option 1

```
Sprint 1 (13.11.2013 - 27.11.2013)
Sprint 2 (27.11.2013 - 11.12.2013)
Christmas vacation
Sprint 3 (7.1.2014 - 20.1.2014)
Sprint 4 (20.1.2014 - 3.2.2014)
Sprint 5 (3.2.2014 - 24.2.2014)
Sprint 6 (24.2.2014 - 10.3.2014)
Sprint 7 (10.3.2014 - 24.3.2014)
Sprint 8 (24.3.2014 - 9.4.2014)
```

Option 2

```
Sprint 1 (13.11.2013 - 27.11.2013)
Sprint 2 (27.11.2013 - 11.12.2013)
Christmas vacation
Sprint 3 (7.1.2014 - 16.1.2014)
Sprint 4 (16.1.2014 - 30.1.2014)
Sprint 5 (30.1.2014 - 13.2.2014)
Sprint 6 (13.2.2014 - 27.2.2014) (An exam week)
Sprint 7 (27.2.2014 - 13.3.2014)
Sprint 8 (13.3.2014 - 27.3.2014)
Sprint 9 (27.3.2014 - 9.4.2014)
```

Option 3

```
Sprint 1 (13.11.2013 - 27.11.2013)
Sprint 2 (27.11.2013 - 11.12.2013)
Christmas vacation
Sprint 3 (7.1.2014 - 16.1.2014)
Sprint 4 (16.1.2014 - 30.1.2014)
Sprint 5 (30.1.2014 - 13.2.2014)
Sprint 6 (13.2.2014 - 6.3.2014) (3 weeks because of an exam week)
Sprint 7 (6.3.2014 - 20.3.2014)
Sprint 8 (20.3.2014 - 3.4.2014)
Sprint 9 (3.4.2014 - 9.4.2014) (Wrapping up)
```

Table 7 presents project related dates that are important to remember. It does not contain upcoming sprint changes or other times relating to sprints as our schedule is not confirmed yet.

Date	Time	Event
13.11.2013		Sprint 1 starts
26.11.2013	klo 14:15-16	EES: Project managers
26.11.2013	klo 16:15-18	EES: QA managers - req. engineering
27.11.2013		Sprint 2 starts
09.12.2013	klo 13	DL: All documents
11.12.2013	klo 11-12	Project review (PP)
12.12.2013 - 06.01.2014		Christmas vacation
08.01.2014 - 20.01.2014		PM, QA, Customer and Mentor in Canada
28.01.2013	klo 14:15-16	EES: Architects/Developers
28.01.2013	klo 16:15-18	EES: QA managers - quality
29.01.2014	klo 16-18	EES: Project managers
17.02.2014	klo 13	DL: All documents
19.02.2014	klo 11-12	Project review (I1)
31.03.2014	klo 13:00	DL: Delivery of the finalized system to the peer group
01.04.2014	klo 14:15-16	EES: Architects/Developers
01.04.2014	klo 16.15-18	EES: QA managers
02.04.2014	klo 16-18	EES: Project managers
03.04.2014	klo 13	DL: Delivery of the peer testing results to the peer group
07.04.2014	klo 13	DL: Delivery of all documents
08.04.2014 - 09.04.2014		Project review (I2)

Table 7: Important dates

6.2 Sprint 1 Plan

Sprint goal
Everyone understands the domain of the project and customer collaboration has been started. ✓

General goals

- To understand Agilefant's vision ✓

- To have the main requirements from the Customer ✓

- To understand the used process ✓

- To understand the domain ✓

- To have a draft of UI using wireframes ✓

- To know required technologies ✓

- To know tools that are going to be used ✓

- Working place ready at A243 with a couple of work stations ✓

6.3 Sprint 2 Plan

Sprint goal
To develop version 0.1 with a simple task queue view.

General goals

- To have the development environment set up to everyone ✓

- To have everyone working with the code ✓

- To have the code base ready

- To have the high-level architecture design ready

- To have the wireframes ready ✓

- To identify risks in the project ✓

- To have an initial architecture ready ✓

Deliverables

- Project plan (no QA plan) ✓

- Progress report slides ✓

- Contract (one per a group) ✓

- Requirements document (except details of requirements) ✓

6.4 Sprint 3 Plan

General goals

- To get to know the Canadian students

- To familiarize Canadian students to the practices and processes

- To form three distributed teams

Deliverables of Implementation 1 phase

- Project plan

- Progress report

- Requirements document

- Technical specification (at least the general architecture)

- QA plan

- Test cases and test log

Deliverables of Implementation 2 phase

- Project plan

- Project review slides

- Final report

- Requirements document

- Technical specification

- User's manual

- Peer test materials to peer group

- Peer test results to peer group

- QA plan

- Test cases and test log

- Peer test session charters with logs (own and peer group's)

7 Risk log

ID	Risk	Prob.	Sev.	Effects	Controlling actions	Resp.
1	A developer quits in the middle of the project.	2	3	Some knowledge is lost. Project scope must be decreased.	Taking care of good team spirit. Using pair programming.	PM / team
2	Adapting with Scrum practices is harder than expected.	2	2	Productivity is lower than assumed and stories cannot be finished on time.	Providing enough training.	PM
3	The team could not build a code base that is wide enough to divide development tasks among three teams on January.	3	3	All developers cannot start development right away, but need to wait until the code base is ready. The project scope must be decreased.	Architectural design needs to be started as soon as possible. Team members should start coding small features as soon as possible to get familiar with technologies.	AR
4	The customer has to leave the CS-building on January, 2014.	2	3	Getting feedback takes longer and the amount of face-to-face meetings decreases.	To generate a backup plan of how stay contacted with the customer.	PM / Customer
5	Merge conflicts in Git when all have privileges to push code to master.	3	3	Lots of effort have to put to resolve the problems.	Git training should be provided to everyone if needed.	PM
6	Non-experienced users mess the Git repository.	3	3	Lots of effort have to put to resolve the problems.	Git training should be provided to everyone if needed.	PM
7	Major changes in requirements.	1	3	The outcome of the project will suffer.	We should collaborate actively with the customer to stay informed about requirements.	Customer / teams
8	Communication with off-site team members won't work as supposed.	2	3	Stories are not finished on time. People won't know what others are doing.	Scrum master should observe that good communication practices are followed.	SM
9	A team divides to sub-teams.	3	3	Team's efficiency decreases and the team spirit suffers.	Scrum master should observe procedures within a team.	SM
10	REST API is not working properly.	2	3	All data is not available. Effort should be allocated to develop the REST API.	Customer should put effort to ensure that the REST API is working properly.	Customer, AR
11	Lack of proper testing devices.	3	2	Correct behaviour of the application cannot be tested using wide range of real devices.	Using our own devices and different emulators.	Customer
12	The Finns' hours run out before the end of the course.	3	3	Cooperation with Canadian students is harder.	Hour usage need to be followed. Developers with lower amount of credits work less during the fall.	PM
13	Chosen technologies are not suitable for our needs.	1	3	Major changes need to be done and project scope will be decreased.	Architecture should be well designed together with the customer at the beginning of the project.	AR

Table 8: A risk log (Probability: 1=lowest, 3=highest, Severity: 1= lowest, 3=highest)

7.1 Materialized risks

Risk 1

One developer quit during the first sprint as he found out he wouldn't be able to participate in the project as required. We discussed about it with the mentor and the team, and we decided that he will take the course later again.

8 References

[1] Project Management Guidelines, "http://www.soberit.hut.fi/T-76.4115/13-14/instructions/project_management.html#Project_review"

[2] Project Management Guidelines, "http://www.soberit.hut.fi/T-76.4115/13-14/instructions/template/project_plan.html"

[3] Scrum Guide, "https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf"