

Project Plan

Mobilefant

December 5, 2013

Contents

1	Introduction	3
1.1	Vision	3
2	Stakeholders and staffing	3
2.1	The Team	4
2.2	Mentor	5
2.3	Customer	5
3	The Goals	5
3.1	Project goals	5
3.2	Personal goals	6
4	Resources	6
4.1	Personnel	6
4.2	Material	6
5	Work practices	7
5.1	Practices	7
5.1.1	Iterative development	7
5.1.2	Sprint planning	8
5.1.3	Documenting	10
5.1.4	Risk management	10
5.1.5	Time tracking	10
5.1.6	Communication	12
5.1.7	Defect tracking	12
5.1.8	Version control	12
5.1.9	Process improvement	13
5.1.10	Requirement engineering	13
5.1.11	Design	13
6	Phasing	13
6.1	Schedule	14
6.2	Sprint 1 Plan	14
6.3	Sprint 2 Plan	14
7	Risk log	15
7.1	Materialized risks	15
8	References	15

1 Introduction

Agilefant is an open source tool for task and requirement management for agile software development. It is provided as an open-source version and a hosted version. The hosted version comprises more and better 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. Agilefant's main competitors are already providing mobile applications, so it is crucial to Agilefant to response for this. Therefore, the goal of our team is to develop a mobile application that works along the hosted version of Agilefant and can be used on both smart phones and tables.

1.1 Vision

Agilefant's vision is to become the leading provider of agile backlog management tools.

2 Stakeholders and staffing

The project contains several stakeholders, which are presented in Figure 1. The stakeholders are divided into four groups: the customer (Agilefant), the student group, the teaching personnel in Aalto University, and teaching personnel in University of Victoria. Arrows in the figure present the direction of main communication.

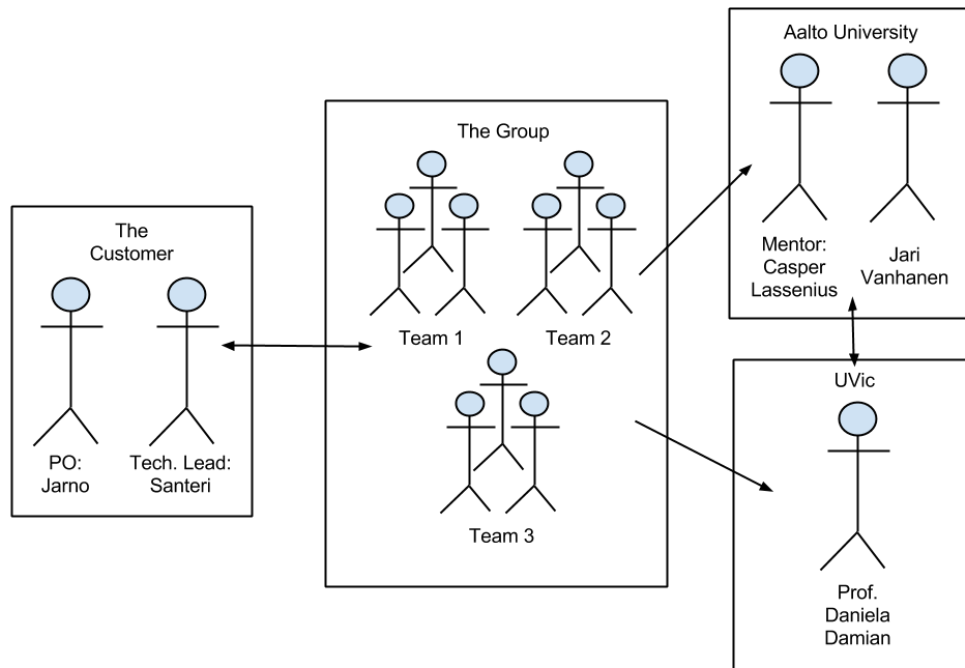


Figure 1: Stakeholders of the project

2.1 The Team

Here we are only listing the role, name, email, responsibilities and an assistant role of each team member. We have a document with everyone's personal informations such as email, phone number and Github name, but we won't publish those informations excluding email.

The group's email is mobilefant#agilefant.org.

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

- Project Manager = PM
- Architect = AR
- Quality Assurance = QA
- Requirements Engineering = RE
- Developer = Dev

Role	Name	Email	Responsibilities	Assistant role
PM, RE	Benjamin Behm	benjamin.behm#aalto.fi	Organizing the work, removing impediments, documenting, process supervising, eliciting requirements, coding	-
AR, RE	Harri Lampi	harri.lampi#aalto.fi	Architectural design, eliciting requirements	-
QA, RE	Matias Kuusela	matias.kuusela#aalto.fi	QA	
Dev	Miro Vilkki	miro.vilkki#aalto.fi		
Dev	Rolle Saarinen	rolle.saarinen#aalto.fi		
Dev	Janne Gröndahl	janne.grondahl#aalto.fi		
Dev	Janne Kajovuori	janne.kajovuori#aalto.fi		
Dev	Joakim Kronqvist	joakim.kronqvist#aalto.fi		

Table 1: The team

NB! Each developer should act as an assistant to some of the SE experts in order to get a broader view of the project.

2.2 Mentor

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

Table 2: Mentor

2.3 Customer

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

Table 3: Customer representatives

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. In order to support these learning goals, tasks are

tried to assign to people so that the course help them to learn what they want to learn. Other high level goals are identified, and these are presented in Table 4.

#	Goal	Verification Criteria
1	To produce high customer satisfaction	Customer's personal opinion about the delivered product.
2	To build 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 get grade 5	The grade will be visible in transcript of records or the course personnel has verified the grade.
5	To win the quality award	Our group has selected as the best group at the end of the course.

Table 4: Project goals in the priority order

3.2 Personal goals

As we are here learning new things, we should focus to learn things we are interested in. Thus, it is important that everyone tells their interests aloud and points out what they would like to learn during this course.

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 takes some responsibility of that manager role.

Personal learning goals can be found in Google Docs: [Learning Goals](#)

4 Resources

4.1 Personnel

Each member must invest “credits * 27 hours - 15 hours in the project”.

[Link](#) to the time allocation page. Everyone should mark how much time he/she is going to use per a week to the table.

4.2 Material

We need mobile phones to test the application. The customer has promised to deliver some test phones, but a wide range of different phones with different platforms cannot be guaranteed. In this reason we will use our own mobile phones for testing.

The CSE department borrows three desktop computers to our group with Ubuntu 12.04 installed. These computers have been set up to our team room A243.

The room (A243) will be shared with an another project group (#15 - TrafficSense) so we need to schedule the usage of the room with them. The idea will be that both

teams will have specific days and hours the room is exclusively reserved for them. At other times, everyone could use the room.

A development environment can be downloaded from Internet if needed. Eclipse is an open-source and free to download, and the project manager has a JetBrains Classroom License, so that IntelliJ IDEA Ultimate can be used during the course.

5 Work practices

In this section we describe what working practices we have planned to use on 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. Its focus is to provide guidelines that enable flexible way of working and allow the development team to react to changes in requirements faster than traditional development frameworks.

The course is divided into three phases (Planning, Implementation 1 and Implementation 2). We don't emphasize these phases as we have planned to have two sprints before the Christmas and seven sprints after the Christmas. Because of the exam week on week 8, we have one sprint of three weeks whereas the other sprints are two weeks.

We will be following the scrum practices meaning that we will keep sprint planning, sprint review and sprint retrospective sessions in every sprint.

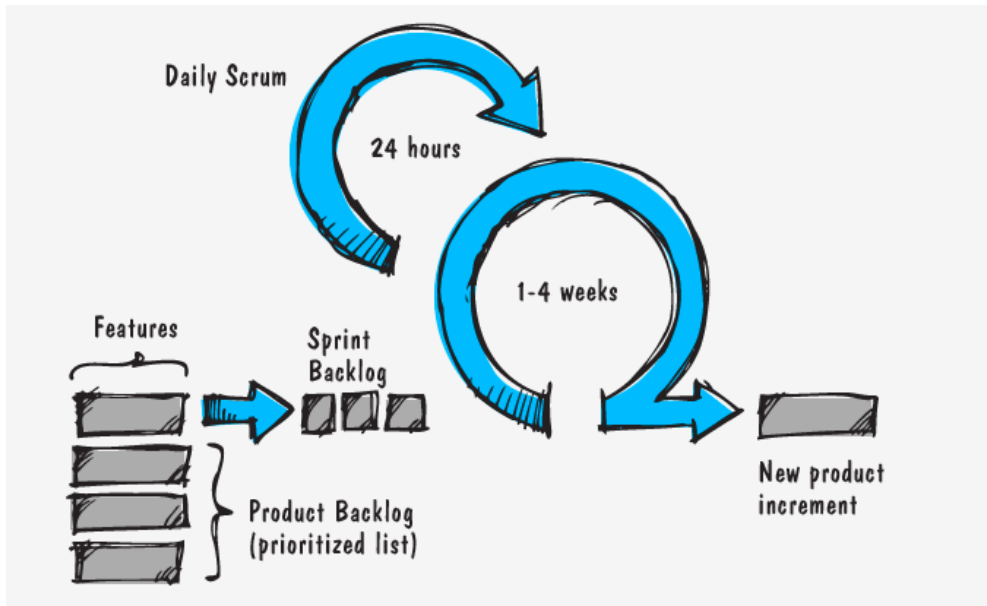


Figure 2: Scrum process

5.1.2 Sprint planning

Sprint planning session will be divided into two parts. The content of the sprint planning is presented in Table 5. The project management has 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 Canadiens are located in their class room in Vancouver, Canada.

Part	Duration	Description	Participants
1	1h	The product owner presents the prioritized product backlog, so that the teams would understand what should be done during a following sprint. The product owner is there for answering any questions the teams would like to ask relating to the user stories and tasks. Then the teams select items from the product backlog to the sprint backlog based on their knowledge of how much work they are capable of doing during a sprint. Sprint goal is agreed in this part.	Product owner, members of each team
2	2h	Teams are separated 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 was assigned to. In this meeting, the team can start design the system so that they are able to convert the backlog items into a working software increment.	Team members

Table 5: The content of a sprint planning

In a sprint planning session (excluding sprint 1), stories are estimated using story points following the pattern of fibonacci numbers. Story points are not directly relating to possible time needed to finish a story but rather describe their

Possible story points are listed below with approximate time:

1
2
3
5
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.

The customer said we should not have larger stories than 10 points.

5.1.3 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 group's web page is located in [Github](#). That works as a base for links to actual documents.

The SE trio is responsible for writing mandatory course documentation. Developers can help to write documents based on their interests. Their effort is preferable at least when planning the architecture and eliciting requirements from the customer.

5.1.4 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 we identified several potential risks. When these risks were identified, we estimated the probability and severity of each risk and discussed a little bit how to minimize the occurrence of these risks.

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

5.1.5 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 to the 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.

As the group is logging spent effort to the Agilefant, the customer is able to follow whether the group is working as promised.

#19531: Kickoff Meeting

Done Benjamin 15h Edit

Labels

This story has no labels

Parent story

(none)

Show hierarchy

Reference ID

story:195313

Description

(empty)

Tasks

Create task

#	Name	State	Responsibles	Left	Spent	Edit
+	First meeting	Done	Aleksi, Benjamin, Harri, JanneG, JanneK, Joakim, Matias, Miro, Rolle	—	15h	Edit

Figure 3: Story and task with spent effort

Spent effort

Effort logger on this element

Date	User	Spent	Left	Comment	
2013-11-14 19:02	Benjamin	0.5h	0		Save
2013-11-14 12:18	Aleksi Hoffman	1.7h	—		Delete
2013-11-14 12:18	Benjamin Behm	1.7h	—		Delete
2013-11-14 12:18	Harri Lampi	1.7h	—		Delete
2013-11-14 12:18	Matias Kuusela	1.7h	—		Delete
2013-11-14 12:18	Miro Vilkki	1.7h	—		Delete
2013-11-14 12:18	Rolle Saarinen	1.7h	—		Delete

My spent effort

11.11.	12.11.	13.11.	14.11.	15.11.	16.11.	17.11.	18.11.	Total	
3h 30min	—	6h	5h 40min	—	—	—	—	15h 10min	Week 46 (11.11.)
									Current week

Close

Figure 4: Log spent effort

When the course is over, credits will be given based on the hours logged to the Agilefant (+ hours spent on lectures). The view shown in Figure 5 can be found in 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 5: Total used hours

5.1.6 Communication

Team will keep a daily standup meeting every time they gather together to work. The daily standup 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 meeting, what they will do before the next meeting, and what obstacles are in the way.

The product manager will propose if the team could use Flowdock as the main communication tool. Aalto provides 180 days license for that.

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

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

5.1.7 Defect tracking

For defect tracking we are using Github's **Issues** functionality.

5.1.8 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.

We have a private repository under Soberit organization. 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 Github in order not to mess the whole repository as everyone has privileges to push anything into master.

We have defined a general git workflow. It can be found [here](#).

5.1.9 Process improvement

A retrospective is arranged at the end of each sprint. The goal of having regular retrospectives is to improve the process and avoid roadblocks in development process.

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 root cause of problems. In this way, these meetings can be more productive and less time will be wasted.

In general the ideas that are generated during a retrospective meeting are divided into two categories: what went well and what could be improved.

The retrospective meeting is time-boxed to one hour (this depends on how much time Canadians have). 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 meetings.

The retrospective contains three phases:

1. First, everyone enters problems to ARCA-tool.
2. Second, the entered problems are discussed together.
3. Third, everyone enters underlying causes to the problems identified earlier.
4. Fourth, causes are discussed and additional causes are added if needed.
5. Fifth, findings are summarized at the end of the meeting.

5.1.10 Requirement engineering

Requirement eliciting is up to the SE trio, but other team members can also participate in the eliciting process.

Requirements are collected to Agilefant.

The customer is responsible for prioritizing the product backlog that is located in Agilefant.

Requirements should be presented as a format of a user story. This format helps everyone to capture the who, what and why of a requirement. The template for a user story is following:

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.

5.1.11 Design

6 Phasing

Tasks are not listed in this project plan, as they are listed and maintained in [Agilefant](#).

6.1 Schedule

Sprint 1 (13.11.2013 - 27.11.2013)

Sprint 2 (27.11.2013 - 11.12.2013)

Christmas vacation

Sprint 3 (7.1.2014 -

Sprint 4

Sprint 5

Sprint 6

Sprint 7

Sprint 8 (X.X.2014 - 9.4.2014)

6.2 Sprint 1 Plan

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

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

Deliverables:

- Project plan (no QA plan)
- Progress report slides
- Contract (one per a group)
- Requirements document (except details of requirements)

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 / All
2	Adapting with Scrum practices is harder than expected.			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 asap. Team members should start coding small features asap 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 when all have privileges to push stuff 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.	Product owner, 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.	Scrum master
9	A team divides to sub-teams	3	3	Team efficiently decreases and the team spirit suffers	Scrum master should observe procedures within a team	Scrum master

Table 6: A risk log (Probability: 1=lowest, 3=highest, Severity: 1= lowest, 3=highest)

7.1 Materialized risks

One developer quit during the first sprint as he found out he wouldn't be able to participate in the project as needed.

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"