

Project Plan

Mobilefant

November 16, 2013

Contents

1	Introduction	3
1.1	Vision	3
2	Stakeholders and staffing	3
2.1	The Team	3
2.2	Mentor	4
2.3	Customer	4
3	The Goals	4
3.1	Project goals	4
3.2	Personal goals	5
4	Resources	5
4.1	Personnel	5
4.2	Material	5
5	Work practices	6
5.1	Practices	6
5.1.1	Iterative development	6
5.1.2	Sprint planning	7
5.1.3	Documenting	8
5.1.4	Risk management	8
5.1.5	Time tracking	8
5.1.6	Communication	10
5.1.7	Defect tracking	10
5.1.8	Version control	10
5.1.9	Process improvement	10
5.1.10	Requirement engineering	11
5.1.11	Design	11
5.1.12	Practice X	11
6	Phasing	11
6.1	Schedule	11
6.2	Sprint 1 Plan	11
6.3	Sprint 2 Plan	12
7	Risk log	12
8	References	12

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 group's web page is located in [Studentwiki](#).

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

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.

We have several roles and one person can have several roles. 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, AR	Aleksi Hoffman	aleksi.hoffman#aalto.fi		
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 to 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 mobile application for Agilefant that contains the main functionalities of its cloud version and fulfill customer's vision of the product. 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. The CSE department can borrow desktop computers to our group with virtual machines installed. These computers need to be 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). Planning phase has two 2-weeks sprints. Implementation 1 and Implementation 2 have both 3 sprints, two 2-weeks sprints and one 3-weeks sprint. It needs to be remembered that there are exam weeks in Finland, so those affect the times when students are willing to work for the project.

A sprint contains four phases: sprint planning, development, demo, and retrospective.

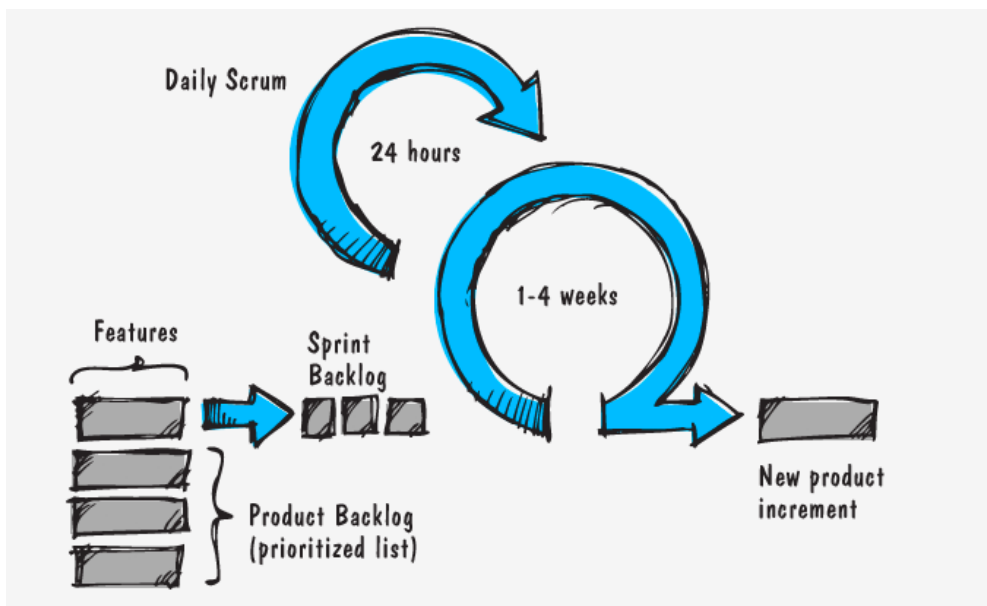


Figure 1: Scrum process

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.

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, team members
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 fibonacci numbers. Story points will be given based on people's opinion of how much time it requires to finish the story. Possible story points are listed below with approximate time:

- 1:** without a break
- 2:** half a day
- 3:** a day (= full work day for a pair)
- 5:** two work days
- 10:** five days

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.

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 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

5.1.5 Time tracking

The groups' 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 and those also tell whether estimated hours are correlating with actual hours. That helps 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.

The screenshot shows the Agilefant interface for a story titled "#19531: Kickoff Meeting". The story is in a "Done" state, assigned to "Benjamin", and has an estimated effort of "15h". The story details include: Labels (This story has no labels), Parent story (none), Reference ID (story:195313), and Description (empty). Below the story details is a "Tasks" section with a "Create task" button. The tasks table has columns: #, Name, State, Responsibles, Left, Spent, and Edit. There is one task listed: "#1 First meeting", which is in a "Done" state, assigned to "Aleksi, Benjamin, Harri, JanneG, JanneK, Joakim, Matias, Miro, Rolle", with "Left" as "—" and "Spent" as "15h".

#	Name	State	Responsibles	Left	Spent	Edit
+	#1 First meeting	Done	Aleksi, Benjamin, Harri, JanneG, JanneK, Joakim, Matias, Miro, Rolle	—	15h	Edit

Figure 2: 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 Viikki	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	<div>Week 46 (11.11.)</div> <div>Current week</div>

Close

Figure 3: 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 4 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 4: 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

Agilefant could be used

5.1.8 Version control

Git and Github will be used for version control.

TODO: How to use it when 3 teams? Check options from [here](#).

5.1.9 Process improvement

A retro will be arranged at the end of each sprint. There will be three phases:

1. First, we will go through impediments from the previous retro and check if the impediments has been fixed.

2. Second, each team member will write down aspects that has worked well and which might need some attention.
3. Third, these will be collected and written to Excel and everyone should explain what they wrote.

5.1.10 Requirement engineering

Agilefant will be used for gathering requirements from customer and maintaining the backlog.

5.1.11 Design

5.1.12 Practice X

6 Phasing

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

6.1 Schedule

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 the architecture design
- To have a draft of UI using wireframes
- To know required technologies
- To know used and required tools
- Working place ready at A243

6.3 Sprint 2 Plan

Goals:

- Set up the development environment
- Get touch with the code
- To have the code base ready
- To have the architecture design ready

Deliverables:

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

7 Risk log

ID	Risk	Prob.	Sev.	Effects	Controlling actions	Responsible
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.	The team / project manager

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

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”