Mobilefant Project Plan

The team

November 14, 2013

Abstract

The abstract text goes here.

Chapter 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.

Currently Agilefant can be used on a laptop / computer, but it is not responsible to be used on mobile devices. The customer wishes that the users of Agilefant could use its the most important functions using their mobile phones and tablets. The goal of our team is to develop a mobile application that works along the hosted version of Agilefant and can be used on smart phones.

Chapter 2

Stakeholders and staffing

2.1 The Team

Role	Name	Responsibilities	Assistant role
Project Manager	Benjamin Behm	Organizing the work, removing impediments, documenting, process supervising, coding	-
Architect	Harri Lampi	Architectural design	-
Quality Assurance	Matias Kuusela	QA	-
Developer	Aleksi Hoffman		
Developer	Miro Vilkki		
Developer	Rolle Saarinen		
Developer	Janne Grndahl		
Developer	Janne Kajovuori		
Developer	Joakim Kronqvist		

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
Mentor	Casper Lassenius

2.3 Customer

Role	Name
Product owner	Jarno Vahaniitty
Tech. Lead	Santeri Korri

Chapter 3

The Goals

3.1 Project goals

The main goal is to develop an mobile application for Agilefant that contains the main functionalities of its cloud version.

Table 1: Project goals in the priority order

Goal	Verification Criteria
1	-
2	-

3.2 Personal goals

Personal learning goals can be found in Google Docs: Learning Goals

Chapter 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

Everyone should have a computer. We also need mobile phones to test the application. It could be possible to get desktop computers (or extra screens) if we get a own 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 JetBrain's Classroom License, so that IntelliJ IDEA Ultimate can be used during the course.

Chapter 5

Work practices

5.1 Practices

5.1.1 Iterative development

Development will be divided into several sprints so that after every sprint we would have an improved version of the application ready to release.

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

5.1.2 Sprint planning

Sprint planning session will be divided into two parts. The content of the sprint planning is presented in Table 3.

Stories will be estimated based on fibonacci numbers. Story points will be given based on peoples opinion of how much time it requires to finish the story. Possible story points are listed below: 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 splitted to smaller stories so that it can be finished during the sprint.

Table 3: The content of sprint planning Part Time 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 splitted 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

5.1.3   Documenting

5.1.4   Risk management

5.1.5   Time tracking

Time tracking will be done using Agilefant. Everyone should log used hours directly to the story or task the person has been doing.

When the course is over, credits will be given based on the hours logged to the Agilefant (excluding lectures).

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.

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 do 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

Chapter 6

Phasing

Chapter 7

Risk log

Table 4: A risk log (Probability: 1=lowest, 3=highest, Severity: 1= lowest, 3=highest).

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

Bibliography

[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"