Project plan+study diary

SpaceHunt

version 1.5

|  |  |  |
| --- | --- | --- |
| TUT | Pervasive Computing | TIE-21106 Software Engineering Methodology |
| Author: TheGroup | | Printed: |
| Distribution: Anna Vaňková, Miloš Švaňa, Nejc Maček, Wladimir Hofmann, Ali Doruk Gezici | | |
|  | | |
|  | | |
|  | | |
| Document status: draft | | Modified: 20.01.2018 14:55 |

VersioN historY

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Authors | Explanation (modifications) |
| 1.0  1.1  1.2  1.3  1.4  1.5 | 28.01.2014  11.02.2014  18.01.2015  26.1.2015  16.01.2017  08.01.2018 | Marko L.  Marko L.  Tensu  Marko L.  Kari S.  Farshad A. | Initial version  Deleted finnish text  Sections 1.4.x, cosmetic tuning  Final toucher  Adaptation for 2017 needs  Adaptation for 2018 needs |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table Of CONTENTS

1. PROJECT RESOURCES 4

1.1 Personnel 4

1.2 Process description 4

1.3 Tools and technologies 4

1.4 Sprint backlogs 5

1.4.1 Sprint 1 5

2. StUDY DIARY 6

2.1 Sprint 1 (every sprint as a section) 6

2.1.1 Subsections 6

2.2 Sprint 2 6

3. RISK MANAGEMENT PLAN 7

3.1 [example] Personnel risks 8

3.1.1 [example] Risk P1: short term absence of one person 8

3.2 [example] Technology risks 9

3.2.1 [example] Risk T1: hard disk failure 9

# introduction

# PROJECT RESOURCES

## Personnel

### Anna Vanková

* Role: Scrum Master
* Contact information:
  + [anna.vankova@student.tut.fi](mailto:anna.vankova@student.tut.fi)
  + WhatsApp and phone: +420 730 952 330
  + Skype: aniicka94
  + FB: Anna Vaňková
* Previous expirience
  + basics of Java
  + experiences with project management
  + design and implementation of project templates for Erasmus+ project – bachelor thesis
  + project manager in Erasmus+ mobilities
  + MS Project
* Specific skills
* Fields of interest
  + Project Management
  + Teamwork
* Absence: 8.2. – 12.2., 30.3. – 4.4.
* Working hours: 7 hours per week

### Milos Svana

* Role: Developer
* Contact information:
  + [milos.svana@student.tut.fi](mailto:milos.svana@student.tut.fi),
  + WhatsApp and Slovak phone: +421 918 075 608
  + Finnish phone: +358 449 531 779
  + Skype: xauder1
  + LinkedIn: <https://goo.gl/zsAZLJ>
* Previous expirience:
  + many group projects during bachelor’s degree studies
  + developer for a small book social network project
  + developer and data analyst for heavy machinery business intelligence company
* Specific skills:
  + Python (Numpy, Sklearn),
  + advanced SQL,
  + PHP,
  + Javascript,
  + C,
  + C++,
  + basics of android applications development in Java
* Fields of interest: Data science, behavioral economy
* Absence: none
* Working hours: 7 hours per week, mostly on Friday

### Wladimir Hofmann

* Role: Product Owner, Developer
* Contact information:
  + Mail: [wladimir.hofmann@student.tut.fi](mailto:wladimir.hofmann@student.tut.fi)
  + Skype/FB: Fladdi.Mir
  + Phone/Whatsapp: +49 176 983 538 35
* Previous experience
  + Material flow simulation
  + Programming (C#, Java, C++ - Basic level)
* Specific skills: we will see :D
* Fields of interest: Programming, Teamwork, & Fun
* Absences: 08.-10.03.
* Working hours: 7h/week

### Nejc Macek

* Role: Developer
* Contact information:
  + [macek@student.tut.fi](mailto:macek@student.tut.fi)
  + [macek.nejc@gmail.com](mailto:macek.nejc@gmail.com)
  + [nejc.macek@student.um.si](mailto:nejc.macek@student.um.si)
  + WhatsApp and phone: +386 31 499 098
  + Skype: [nejcmacek](callto:nejcmacek)
  + Facebook: [Nejc Maček](https://www.facebook.com/macek.nejc)
  + LinkedIn: [Nejc Maček](https://www.linkedin.com/in/nejc-macek/)
  + Website: [nejo.si](https://nejo.si/)
* Previous experience:
  + Employment at PovioLabs
  + Employment at Agitavit
  + Voluntary services at Gimnazija Celje – Center
  + University projects:
    - c++ Variadic Templates research
    - [OpenScience Alternative Frontend](https://openscience.si/iskanjeNejcMacek/)
  + [Many personal projects](https://apps.nejo.si/)
* Specific skills:
  + Web develpment (JS, TS, CSS, HTML, React, Angular, jQuery)
  + Node.js develpment
  + C#, WinForms, .NET, .NET Core, ASP.NET
  + Java
  + c++
  + See also: [nejo.si/skills](https://nejo.si/skills/)
* Fields of interest: web development, web applications, frontend development
* Absence: none
* Working hours: 7 hours per week

### Ali Doruk Gezici

* Role: Developer
* Contact information:
  + [ali.gezici@student.tut.fi](mailto:ali.gezici@student.tut.fi)
  + [doruk@gezici.me](mailto:doruk@gezici.me)
  + WhatsApp and Phone: +358 41 705 6467
  + Skype: doruk.gezici
  + Facebook: Doruk Gezici
  + LinkedIn: Doruk Gezici
  + Website: [dorukgezici.com](http://dorukgezici.com/)
* Previous experience:
  + iOS Developer at Kolay Randevu
  + IoT & Mobile Developer at ARDIC R&D
  + IT Coordinator at EESTEC LC Istanbul
  + IoT Line Fair ’17 Coordinator at EESTEC LC Istanbul
  + Django Developer at ITUscheduler
* Specific skills:
  + Web Development (Python/Django, HTML, CSS, Bootstrap, JS)
  + Python, C++, Software Development
  + Swift, iOS Mobile Application Development
  + Linux SysAdmin
  + Teamwork, Coordinator
* Fields of interest:
  + Entrepreneurship
  + Mobile & Back-End Web Development
  + Linux SysAdmin & Free Software
* Absence: None
* Working hours: 7 hours per week

## Process description

### Project roadmap

We will measure success by gathering feedback from our friends. After every sprint we will send the game to our friends, they will test it and give us a feedback.

#### Sprint 1

Environment setup and basic architecture. Our goal (milestone) will be that player can move in the environment.

#### Sprint 2

Implementing 2 levels of the game. We will have a playable game with 2 levels.

#### Sprint 3

Implementing another 2 levels of the game. We will have a playable game with 4 levels.

#### Sprint 4

Implementing score system, storytelling, testing, fixing bugs and improving graphics.

### Communication

We will have weekly meetings each Friday from 2pm to 4pm. After this meeting we will have an optional coding session. The first one will be held on 26.1. with the aim of setting up the develoment envirnoment and preparing the application structure. We plan to use WhatsApp and Skype for remote communication.

### Responsibilities

Implementation of individual user requirements/stories will be delegated on the start of each sprint to one of the developers.

#### Anna

Storytelling, project management – Agilefant

#### Milos

Code review, accepting/declining merge requests to master branch, implementation of user requirements.

#### Ali Doruk

Development Environment, Server Management, Implementation of User Stories

#### Wladimir

Implementation of user stories

#### Nejc

Software and project architecture. Implementation of user stories.

*Describe your (preliminary) roadmap for the project (including milestones). What are your project goals and success criteria?* ***How do you define success and measure it?***

*How do you run your project? For example, do you communicate by having weekly meetings, sending e-mails, IRC channels and other means of communication? Who is responsible for what? Try to keep responsibilities clear, so that there will not be situations where magical “someone” (Sam Body or some other Body brother) will take of care of something crucial. You should also be able to commit, so taking a responsibility on something should also mean autonomy to take care of the duty. One cannot be held accountable, if the person did not have the necessary freedom and support. Common working moments (****”coding nights”****) together with the whole groups are highly recommended.*

*Check also the risks in chapter 3, as your process should make you at least robust to the risks. For example, how to avoid impact of absent key person? In addition, it is not enough to be robust, you should also learn. So, how do you get feedback from the team, assistant and so on? Daily Scrums, sprint autopsies and other such events?*

*KEEP THIS UPDATED AS YOU LEARN DURING THE COURSE.*

## Tools and technologies

*What tools and technologies you are utilizing during the project? If there is possibility of version compatibility issues, the version number should be mentioned. If a tool is hard to use, you can assign someone to be the contact or* ***responsible person (“guru”) on the tool****. How do you react if new versions of tools emerge during the project? If you are using version control repository, please describe how to access it. Also AgileFant URL can be documented here.*

Table 1.1: Tools used in the project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Tool** | **Contact person** | **version** |
| Documentation | MS Word (word processing)  [office.microsoft.com](file:///\\intra.tut.fi\..\..\..\..\..\Local%20Settings\Temp\office.microsoft.com) | A.V. | 2010 |
| Communication | WhatsApp  <https://www.whatsapp.com/> | A.V. | 7.22 |
| Skype (internet calls)  <http://www.skype.org> | A.V. | 4.64 |
| Version management | Git  <https://git-scm.com/>  Repository:  <https://course-gitlab.tut.fi/sweng_2018/g10---thegroup> | M.S. | 1.4.6 |
| Development platform | Javascript |  | Chrome 63 runtime or later |
| Project Management | AgileFant  <https://app.agilefant.com/login> | A.V. |  |
| Editor/IDE | Visual Studio Code  <https://code.visualstudio.com/> | N.M. | latest |
| IntelliJ platform  <https://www.jetbrains.com/idea/> | M.S. | 2018.3 or later |

# StUDY DIARY

This chapter holds your journal of lessons learned during the course. That is, **more detailed analysis of previous Sprint’s contents**.

## Sprint 1 (every sprint as a section)

### What went well

### What difficulties you had

### What were the main learnings

### What did you decide to change for the next sprint

## Sprint 2

### What went well

### What difficulties you had

### What were the main learnings

### What did you decide to change for the next sprint

# RISK MANAGEMENT PLAN

Our basic risk-prevention policy is shared code ownership, by which we mean, that everybody should at least at a very basic level know about code written by others. Anybody can view and modify any part of the codebase. This allows others to finish other person’s task in case of sudden absence.

*Consider risks for your project.* ***The most usual risks*** *that will affect projects are due to customer, the team itself and technology.*

*Just listing some risks at the beginning of the project doesn’t help you much… if anything at all.*

*You can try to come up with* ***Plan Bs*** *for the risks. However, remember that the things you won’t expect, will hurt you the most. Thus, focus on the generalities, not on specifics.*

*Try not to underestimate the probability of small and common risks, and not to overestimate the probability of rare and remarkable events. For example, people usually get 1-2 flus during a year, so in 4 months, it is quite probable that one of the team will be sick and may infect others, too. An average flu lasts for more than one week. So, be prepared. On the other hand, getting hurt in traffic so that it will take a week to recover happens to only for 15000 people yearly in Finland (less than 3 permille of population).*

*Be sensitive for weak signals, such as difficulties with new technology or runny noses.*

***You should think of risks in all categories:***

* *customer (ending the project, changing requirements, requirements remain unclear,…)*
* *technologies (hw/sw; hard to acquire, learning new technologies takes time, suitable library is not found,…)*
* *environment (network connections and servers fail,…)*
* *personnel (getting ill, changing jobs, busy with work,…)*
* *project management (bad scheduling, bad communication, forgetting things,…).*

*Usually we calculate risk’s* ***seriousness = severity \* probability****.*

Table 4.1: Project risks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk ID** | **Description** | **Probability** | **Impact** |
| P1 | Short term absence | 2 | 2 |
| P2 | Long term absence | 1 | 3 |
| P3 | Busy with work/school assignment | 3 | 2 |
| T1 | Library dependency issues | 1 | 2 |
| T2 | Learning new technologies issues | 3 | 2 |
| T3 | Framework malfunction | 1 | 3 |
| C1 | Changing requirements | 3 | 1 |
| C2 | Requirements unclear | 3 | 2 |
| E1 | HW failure | 1 | 2 |
| E2 | Internet failure | 1 | 3 |
| E3 | Change of SW environment | 1 | 1 |
| E4 | AgileFant unavailability | 1 | 3 |
| E5 | GitLab unavailability | 1 | 1 |
| PM1 | Forgetting things | 2 | 2 |
| PM2 | Bad communication | 2 | 2 |
| PM3 | Bad scheduling | 3 | 2 |

## [example] Personnel risks

Try to estimate risk probability, use a scale of **1 to 3** (or 1..5) or Small, Medium, Large.

Other criterion will be the impact or severity. So, how the risk will harm you, if realized. Use similar scaling as in probability.

### [example] Risk P1: short term absence of one person

Every major risk in the table will be further elaborated here. Analyze the risks, so that those risks which will hurt you the most are analyzed in more detail than rare and low-impact risks.

However, remember that the low impact risks may have cumulative effects, if they have high probability, and thus occur frequently.

Incorporate your mitigation methods to your process (see 1.2.). However, consider the sensibleness of the measures (risk severity vs. cost). For example, getting a flu shot (vaccination) for everyone in the team would surely be overkill.

**Root cause (source):** description of the risk. A key person will be absent for several days.

**Importance (seriousness):** from the table, basically probability and impact, possibly combined with frequency.

**Avoidance:** if you can lower the probability by preventive means, or even totally suppress (reject) the risk. For example, getting flu shots for everyone will lower the risk of short term sickness.

**Response (prevention):** means to take, if you have weak signals of looming disaster. For example, someone seems to be getting sick or will have a mandatory absence next week, redistribute the work load and share all relevant information, so that the team will be able to carry on.

**Recovery (survival):** the means to take, if other means have failed, and the risk has realized. Plan B. For example, redistribute the workload; focus on the most important features.

## [example] Technology risks

### [example] Risk T1: hard disk failure

**Symptom, early warning sign:** disk makes noise, arbitrary reading errors occur more often than before.

**Source or reason:** hard disk is at the end of its lifespan, or hard hit

on computer while disk was running.

**Probability:** 2 medium (on scale 1-3)

**Seriousness:** 2 medium (on scale 1-3)

**How to avoid:** buy a new disk when starting a project.

**How to prevent:** when first symptoms occur, take additional back-ups and change the disk as soon as possible.

**How to survive:** back-ups, and a replacement disk or whole computer.

## Customer risks

## Environment risks

## Project management risks