Group 9: Project plan & study diary

River Raid 2017

version 1.2

TIE-21106 Software Engineering Methodology	Pervasive Computing	TUT
Printed: 26.01.2014 17:47	Author: Arno Lehtonen	
Distribution: Arno Lehtonen <arno.lehtonen@student.tut.fi>		
Kuanysh Kairbek <kuanysh.kairbek@student.tut.fi>		
Mahammad Imranur Rahman <mohammad.rahman@student.tut.fi>		
Tero Ahtee <tero.ahtee@tut.fi>		
Modified: 12.02.2017	Document status: final	

VERSION HISTORY

Explanation (modifications)	Authors	Date	Version
Initial additions, risks	Arno L.	26.01.2017	1.0
Refinements	Arno L.	29.01.2017	1.1
Sprint 1 Study diary	Kuanysh K.	12.02.2017	1.2
Sprint 2 Study diary	Arno L.	12.03.2017	1.3
Sprint 2 Study diary	Mohammad R.	02.04.2017	1.4

TABLE OF CONTENTS

1. PROJECT RESOURCES

1.1 Personnel

Contribution(hours)	Interests	Skills	Experience	E-mail	Name
5+ hours a week	Web, VR, game development	C/++/#, web full stack	+60 credits software development	arno.lehtonen@student.tut.fi	Arno Lehtonen (Scrum Master)
5 h/week	Full stack	C#, Java	Software Developer	rahman8@student.tut.fi	Mohammad Imranur Rahman
8 h/week	Game development, communications	C/C++	Software Developer	kuanysh.kairbek@student.tut.fi	Kuanysh Kairbek
5+ hours per week	Mobile app and game dev.	web full stack dev.	Web and software development	krishna.bagale@student.tut.fi	Kirshna Bagale

1.2 Process description

The project is going to consist of four sprints. The first milestone is defined as the first crude playable version with all game states included (initialization, game on-going, game over). This milestone is expected to be reached by the end of sprint two. The second milestone is our fully functional end product, which only requires the final testing and finishing touches. This milestone is going to be reached by the end of sprint 3 or midway in sprint 4.

Project goals

The aim of this project is to create a functional end product satisfying the customer requirements. We have defined our goals as follows:

- Fully functional, minimum viable product
  - all 10 customer requirements satisfied

- Some additional features, such as
  - different weapons
  - other pickups
- Polished graphical presentation and UX

Managing the workload

Different modules will be assigned to the members according to their interest and skills. After each sprint we will be meeting for planning the next steps.

Communication

We have set up a Slack, where all group members are present. We're planning on having at least one meeting weekly in person, where everybody is able to discuss the project and the

On top of that, we're planning on having coding nights (or days), where the whole group can get together and advance the project.

Risk management

The risk management is discussed in chapter 3. The most common risks were identified and their impacts on the project were assessed. We're planning on updating the risk list in the end of each sprint.

1.3 Tools and technologies

The tools used in this project are listed below. If there's an update available, we refrain from updating the version unless the features in new version are seriously vital for this project.

Table 1.1: Tools used in the project.

version	Contact person	Tool	Purpose
2017	A.L	Google Docs office.microsoft.com	Documentation
16.9	A.L.	Draw.io (UML tool) http://draw.io	
2.4.1	A.L	Slack	Communication
1.4.6	A.L	Gitlab https://gitlab.rd.tut.fi/sweng-	Version management

		2017/g09---balmora.git	
5.5.0	A.L	Unity	SDK
2017	A.L	Agilefant, www.agilefant.com/TTY-TIE	Management Tool

1.4 Sprint Backlogs

1.4.1 Sprint 1

Developer	Description	Feature
Arno	Player is able to move around, graphics	Controllable player character (Customer req. 3)
Kuanysh	The level scrolls below the player	Scrolling map
Mohammand	Something to shoot at	Dummy enemy class (Customer req. 4)
Arno	Default shooting mechanism, graphics	Player is able to shoot (Customer req. 7)
Arno	Fuel and so forth	Pickups (Customer req. 4)
Mohammad	Game data and states	Gamestate
Kuanysh	Initial Unity project + pushing to Git	The initial project setup

1.4.2 Sprint 2

Developer	Description	Feature
Arno	-	Fuel is used while flying and the jet can be refueled
Kuanysh	The level scrolls below the player	Scrolling map

Mohammad	Inputted when the application starts	Player name (Cust. req. 1)
Kuanysh		River consists of different sections (Customer req. 8)
Arno	Different enemies have different points	The player scores points
Arno	Game data and states	Gamestate
Arno	Enemy class	Enemies & graphics

1.4.2 Sprint 3

Developer	**Description**	**Feature**
Krishna	Printing player's name into a Main Scene of the game	Prompting player name
Krishna	Unity Documentation Reading	-
Mohammad	After prompting the player's name	The game greets the player, tells a compelling background story and the game begins.
Arno	Customer requirement X: A menu.	River consists of different sections (Customer req. 8)
Arno	When the game ends, a scoreboard is shown where the player names are shown with the corresponding score.	Showing the final scoreboard
Kuanysh	The river consists of different sections	Customer req. 8 & 3
Krishna	When the application	Customer req. 1

	starts, it inquires the player's name.	
Mohammad	When a user inputs his name it should be validated	Customer req. 1

2. STUDY DIARY

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

2.1 Sprint 1

2.1.1 What went well

The user requirements are well understood. And also the sprint backlog were well planned. We have divided the tasks according to the skill and the interests and discussed the workflow of the project. We have also prioritized the tasks according to customer value. As we are developing in unity it has many advanced features that makes game development easy and efficient. C# scripting was easy and fast to learn because all of us have experience in programming.

2.1.2 What difficulties you had

In our team not all of us had experience in game development. But one of our team member had his hands on Unity game development. At first some of us had problem in understanding the game physics. Big difficultness we faced is project structure and management of the elements. Git and Unity game development took a while to decide which way we are going to store the project files.

2.1.3 What were the main learnings

Main learning is understanding of Unity3D usage for two of our teammates and cooperating with newbies for more experienced one. We have learnt how to communicate with team members and manage workload. C# scripting is coding part of the project and it was learnt also.

2.1.4 What did you decide to change for the next sprint

Communicating and workload sharing shall be improved on. Moreover, there needs to be a set weekly meeting time, since arranging meetings is a tedious task. And also to add more skill in Unity stack like the features and optimization. Also we should focus on design patterns of game development. So that we use the best practices and organize the project well. We should also improve the user interaction with the game. Which we had overlooked a bit in first sprint.

2.2 Sprint 2

2.2.1 What went well

The game is progressing nicely and has about half of the core customer requirements implemented.

2.2.2 What difficulties you had

One of our group members was sick and we had one extra member joining our group, so there was extra hassle with the work. It seems that communication is kinda hard for some of our group members (or maybe they're busy, I wouldn't know), so Arno ended up doing most of the work this sprint.

2.2.3 What were the main learnings

Everybody needs to be on the map what's gonna happen next and who's gonna do what.

2.2.4 What did you decide to change for the next sprint

As the only person sitting down and actually writing this, I would first want to discuss who is in this and who's not since I'm kinda pissed off at the moment.

2.3 Sprint 3

2.3.1 What went well

Since we are quite familiar with the game already, we have done most of the individual work by ourselves and had some group work, which by the way was the most important learning process. More than half of the requirements were implemented.

2.3.2 What difficulties you had

There was some problem or difficulties regarding the speed smoothing of the objects. And also to figure out the best configuration. And then the problem was solved by discussing what are the possibilities to solve it. And also there were some stories left over from the previous sprint the new group member and the sick one have been doing those. But also there are some small bug fixes should be done

2.3.3 What were the main learnings

The main learning outcome includes designing and developing the remaining part of the game. Learnt that group work generates multiple ideas within the team and can enhance our perception as to how things must be done. Being an interdisciplinary, the newest field of this technology intrigued us, fascinated us. We are glad that this project helped us to incur our knowledge to the next level since the get go.

2.3.4 What did you decide to change for the next sprint

As for the next sprint we will be fixing the bugs and also make the game more user friendly. And there is a new customer requirements which we will be doing on next sprint. Beside that we will add some extra features to the game.

3. RISK MANAGEMENT PLAN

In the risk management plan, we assess the probability and impact of the most common possible risks that could affect the outcome of the project. Risk categories are customer, technology, environment, personnel and project management.

Table 4.1: Project risks.

Impact	Probability	Description	Risk ID
4	2	A member of the group falls ill	P1
5	2	A member of the group quits	P2
3	2	A member of the group slacks	P3
2	2	Hard drive failure	T1
1	3	Technology hard to master	T2
4	2	Project management isn’t up to standards	PM1
1	4	Scrum is not used to its potential	PM2
1	5	Git woes with Unity	E1
2	1	Cloud services not available	E2
5	1	Customer requirements change	C1

3.1 Personnel risks

Risk probability is measured using a scale from **1 to 5**, as well as the impact of the risk. This is multiplied with the impact, on a scale from 1 to 5, which gives an overall seriousness estimate. The greater the number, the greater the risk.

3.1.1 Risk P1: A member of the group falls ill

Root cause (source): A key person will be absent for several days.
Importance (seriousness): 8.
Avoidance: Good hygiene and other preventive measures.
Response (prevention): Redistribute workload, do what you can even though you're ill.
Recovery (survival): The backlog has to be prioritized, features cut out.

3.1.2 Risk P2: A member of the group quits

Root cause (source): A group member is not committed to the project.
Importance (seriousness): 6.
Avoidance: Ensure commitment in the beginning.
Response (prevention): Talk with the member in question before things get out of hand. Communication.
Recovery (survival): Contact course personnel, ask for other help.

3.1.3 Risk P3: A member of the group slacks

Root cause (source): A group member is not committed to the project.
Importance (seriousness): 10.
Avoidance: Ensure commitment in the beginning.
Response (prevention): -
Recovery (survival): Contact course personnel, ask for reduced features. Do all the work with remaining group members.

3.2 Technology risks

Risk probability is measured using a scale from **1 to 5**, as well as the impact of the risk.

3.2.1 Risk T1: Hard drive 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
Seriousness: 2
How to avoid: buy a new disk when starting a project.
How to prevent: Additional backups, buy a new disk.
How to survive: Cloud backups, always push to Git.

3.2.2 Risk T3: Technology hard to master

Symptom, early warning sign: A group member is unable to complete his tasks.
Source or reason: Used technology is too difficult to master in this time frame
Probability: 3
Seriousness: 1
How to avoid: buy a new disk when starting a project.
How to prevent: Additional backups, buy a new disk.
How to survive: Cloud backups, always push to Git.

3.3 Project management risks

3.3.1 Risk PM1: Project management isn't up to standards

Symptom, early warning sign: Communication or workload distribution fails, things aren't getting done.
Source or reason: Lack of communication and involvement.
Probability: 2
Seriousness: 4
How to avoid: Active communication and participation, project management push.
How to prevent: Team members know their roles, communicate.

3.3.2 Risk PM2: Scrum isn't used to its potential

Symptom, early warning sign: Features aren't getting done.
Source or reason: Scrum is a new method for some group members.
Probability: 4
Seriousness: 1
How to avoid: Scrum should be mastered, Scrum master does his job.
How to prevent: Read the scrum material, ensure that scrum master know their job.

3.4 Environment risks

3.4.1 Risk E1: Git woes with Unity

Symptom, early warning sign: Features cannot be pushed because of conflicts.
Source or reason: Unity always modifies some actually untouched files.
Probability: 5
Seriousness: 1
How to avoid: Be careful when adding new things to a commit.
How to prevent: Ensure, that Git knowledge is on a high level.

3.4.2 Risk E2: Cloud services not available

Symptom, early warning sign: 404s, connection problems.
Source or reason: Git/Agilefant servers are down.
Probability: 1
Seriousness: 2
How to avoid: No way to avoid this, should it happen.
How to prevent: Out of our reach.

3.5 Customer risks

3.5.1 Risk C1: Requirements change drastically

Symptom, early warning sign: Features cannot be completed because they change or there are too many of them.
Source or reason: Customer.
Probability: 1
Seriousness: 5
How to avoid: Communication, scrum.
How to prevent: Customer should be informed if a feature cannot be completed.