Group 9: Project plan & study diary

River Raid 2017

version 1.5

TUT	Pervasive Computing	TIE-21106 Software Engineering Methodology
Author: Arno Lehtonen		Printed: 26.01.2014 17:47
Distribution: Arno Lehtonen <arno.lehtonen@student.tut.fi>		
Kuanysh Kairbek <kuanysh.kairbek@student.tut.fi>		
Krishna Bagale <krishna.bagale@student.tut.fi>		
Tero Ahtee <tero.ahtee@tut.fi>		
Document status: final		Modified: 01.05.2017

Version history

Version	Date	Authors	Explanation (modifications)
1.0	26.01.2017	Arno L.	Initial additions, risks
1.1	29.01.2017	Arno L.	Refinements
1.2	12.02.2017	Kuanysh K.	Sprint 1 Study diary
1.3	12.03.2017	Arno L.	Sprint 2 Study diary
1.4	02.04.2017	Kuanysh K.	Sprint 3 Study diary
1.5	01.05.2017	Krishna B.	Sprint 4 Study diary

TABLE OF CONTENTS

1. PROJECT RESOURCES

1.1 Personnel

Name	E-mail	Experience	Skills	Interests	Contribution (hours)
Arno Lehtonen (Scrum Master)	arno.lehtonen@student.tut.fi	+60 credits software development	C/++/#, web full stack	Web, VR, game development	5+ hours a week
Mohammad Imranur Rahman	rahman8@student.tut.fi	Software Developer	C#, Java	Full stack	50
Kuanysh Kairbek	kuanysh.kairbek@student.tut.fi	Software Developer	C/C++	Game development, communications	8 h/week
Kirshna Bagale	krishna.bagale@student.tut.fi	Web and software development	web full stack dev.	Mobile app and game dev.	5+ hours per week

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.

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

management	https://gitlab.rd.tut.fi/sweng-2017/g09---balmora.git		
SDK	Unity	A.L	5.5.0
Management Tool	Agilefant, www.agilefant.com/TTY-TIE	A.L	2017

1.4 Sprint Backlogs

1.4.1 Sprint 1

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

1.4.2 Sprint 2

Feature	**Description**	**Developer**
Fuel is used while flying and the jet can be refueled	-	Arno

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

1.4.3 Sprint 3

Feature	**Description**	**Developer**
Printing player's name into a Main Scene of the game	Request a name and print it on main screen during the game	Krishna
Customer requirement X: A menu.		Arno
Customer requirement 10: When the game ends, a scoreboard is shown where the player names are shown with the corresponding score.	Leader board shown in the end of game	Arno
Customer requirement 8 & 3: The river consists of different sections	Sections of the river divided by bridges	Kuanysh
Customer requirement 2: Tell a compelling background story.	Story of the game	Imran, Krishna

1.4.4 Sprint 4

Feature	**Description**	**Developer**
Extra-feature	Background animation in greetings scene.	Krishna
Customer feature 11: Enemies shoot at the player	Spawning enemies shooting at the player.	Arno
Customer requirement 8 & 3: The river consists of different sections	Sections of the river divided by bridges.	Kuanysh
Customer requirement 2: Tell a compelling background story.	Story of the game.	Krishna

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.

2.4 Sprint 4

2.4.1 What went well

Now we have a better understanding of agilefant technology and unity game development. Remaining features like wall, highscore displaying is developed now and the enemy shooting at the player is also done. We are trying to include some extra features like some animation for the greeting scene.

2.4.2 What difficulties you had

One of the team member was supposed to develop the bridge and was agreed to do that after the meeting of sprint 3. However it turns out that he didn’t contact and didn’t respond to the group and the teacher in anyway whatsoever.

2.4.3 What were the main learnings

The main learning during this sprint was that the awareness of being in multicultural team did lead to the unknown. Again some practices for extra features can be included to the main learnings in this sprint, even though we have problem developing that.

2.4.4 What did you decide to change for the next sprint

As of now, we would like to focus on time management and task separation/delegation. We never know what would happen next when we have members from different cultural background.

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.

Risk ID	Description	Probab ility	Impa ct
P1	A member of the group falls ill	2	4
P2	A member of the group quits	2	5
P3	A member of the group slacks	2	3
T1	Hard drive failure	2	2
T2	Technology hard to master	3	1
PM1	Project management isn’t up to standards	2	4
PM2	Scrum is not used to its potential	4	1

E1	Git woes with Unity	5	1
E2	Cloud services not available	1	2
C1	Customer requirements change	1	5

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): 10.
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): 6.
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 T2: 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: Go through tutorials when starting a project.
How to prevent: Take into account your capabilities to do the task.
How to survive: Read or watch tutorials, examples of the code and ask questions from more experienced teammates.

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.