SOFTWAREDEVELOPMENT PROJECT

Amina Hamza
Linneaus
University
2020-01-30

Contents

1 Revision History

Date	Version	Description	Author
2020-01-30	001	Vision, project plan and risk analysis added. Iteration 1 completed.	Amina Hamza
2020-02-23	002	Iteration 2 and time log was updated.	Amina Hamza
2020-03-09	003	Iteration 3 and time log was updated.	Amina Hamza
2020-03	004	All parts ex was updated, iteration 4 was added.	Amina Hamza

2 General Information

Project Summary	
Project Name	Project ID
Project-Hangman	001
Project Manager	Main Client
Amina Hamza	Computer Science Department
Key Stakeholders	
Customer Developer Tester Promoter End User	
Executive Summary	
This project is going to develop a text-based word guessing game called Hangman that can be played in the console. The player gets to guess one letter at the time and need to guess all the letters before he run out of guesses otherwise he/she loses. The game is developed because there are players looking for a game like this which means there is a market for it. The goal is not to earn a lot of money but to provide a fun playable game for hangman enthusiasts.	

3 | Vision

This game will encapsulate anything around software development, including the iterative process, documentation, and testing. The Hangman game will be improved continuously with each iteration.

In this project a game of hangman is going to be developed. The game should be able to be played in the console window and will therefore be a text-based version where the hanged man will be constructed by available characters on the keyboard.

There should be a regular version where, as in all hangman games, a word will be picked from a predefined list of nouns and the player get to guess letters. In this game though if the player guesses the correct word he gets a point and get to guess on another word and earn a high- score for each correct word. Apart from that there should also be the alternative to play a multiplayer version where one player enters a word and the other player get to guess on the entered word. This will make hangman players looking for a game they can play with their friends interested in this game.

Another addition in this game is that the player should be able to leave the game he is currently playing and then return to it as long as he did not initialize a new single player game. A multiplayer version will be added to enable friends play together.

In the beginning there should be a long list of predefined nouns in English, so the player does not have to enter words by himself, this to increase the game friendliness for single players. But if the player finds a word that he really does not like while playing he should also be able to remove it from the list of nouns.

These additional features could really increase the interest in this hangman game in particular. The ability for the player to adapt the game after his own needs should make more people open to play this particular game.

The game should be developed using a plan-driven project plan and the documentation of the project is very significant.

Reflection

It was a bit difficult to determine how much details should be included in the vision. I know that there should not be anything about the code in the vision, but it was hard to know how many features should be presented and so on. Also, the vision tended to feel a bit short since the hangman game is not that big of a project. The text in the vision feels kind of similar to the one under the headline “scope” in the project plan.

Iteration 4: I Changed the vision to be more inspiring and less describing. It now focuses more on the additions to an ordinary hangman game and why these additions could be useful. Removed some text describing a basic hangman.

1 Project Plan

1.1 Introduction

This project is about creating a hangman game that can be played in the console. The project will be plan-driven beginning with planning, then modeling and finally testing. Features could be added at a later stage, but it is important that the different deadlines are kept.

1.2 Justification

The reason for the project is mainly an educational purpose. To learn more about project managing, learn how to construct a plan that should be followed and divide a bigger task into smaller tasks that can be dealt with step by step. To get used to planning and documentation instead of just diving directly in to coding. It will also be an opportunity to understand how modelling and testing works. During development software and new techniques will be used which also will deepen knowledge in the field.

It should also result in a game friendly to players who want to both be able to play a multiplayer version with their friends but also a single player version where they get random words from a predefined list of nouns.

1.3 Stakeholders

The most important stakeholder in the project is the customer who already has defined the main requirements. The project manager however gets to decide how ambiguous the game should be, how many additional features should be implemented. The knowledge and experience of the project team members also determines the limits of the project.

Customer needs a a complete and good game with no bugs. In this case the customer has predefined the main task and have stated certain deadlines for subtasks and then given the project manager the responsibility to decide on design and implementation.

Developer wants to develop this game and as much as possible make it stable. The knowledge and experience of the programmer will also affect which features is possible and how much time is needed to implement these features.

Testers wants structured code preferably with distinct hierarchy to make automated testing of separate classes possible.

Promoters are going to market and make the game known to as many people and players as possible.

End user – The end user will be able to come with some input about which features would be fun or useful in the game. Towards the end of the development the end user could tell about the experience of playing the game and minor or major sources of irritation that should be handled.

1.4 Resources

Pre-recorded video lectures and Tutoring Sessions provided by the department, Software Engineering 10th ed by Ian Sommerville, and 20hrs a week for the development.

Considering time there is about two weeks for making a project plan, two weeks to design and implement the game and two weeks to test the program. Then there is about another

two weeks to finish the final details of the program, so after eight weeks in total the game should be finished.

1.5 Hard- and Software Requirements

The resource for this project a working computer and the development will be done using JAVA 8, and Eclipse IDE. And to run and play the game, the hardware must have JDK 1.8 at least and minimun memory of 2G.

1.6 Overall Project Schedule

Project Plan - 7th February 2020.

An overall plan of the project containing a vision, project plan, risk analysis and some skeleton code. Detailed plan in iteration 1.

Modelling -20th February 2020.

Use case diagram, fully dressed use cases, state machine diagram, implementation of a somewhat playable version of the game and a class diagram. Most features should be implemented at this stage. Detailed plan in iteration 2.

Testing - 6th March 2020.

A test plan for both manual and automated tests and a test report after performing these tests. The game should be carefully tested, and problems should be fixed. Detailed plan in iteration 3.

Hand-in project – 20/3-2020

The deadline of the entire project. At this point a finished game of hangman should be able to be delivered to the customer. Detailed plan in iteration 4.

1.7 Scope, Constraints and Assumptions

Scope

1. Text-based fashion game.
2. Console Application.
3. Difficulty Level.

Constraints.

1. The game is not a web application.
2. The game will not require User log in.
3. It will have no Sounds.
4. Inexperience on my part, because it's my first project.
5. Not enough time, because I have other courses along this project.

Assumptions.

1. The player should have an overall idea about the game.
2. The player should know how to use console to run the game.
3. The user should have JDK and JRE installed on his/her system.

Reflection

The project plan contains a lot of details about the project like deadlines and which resources are available and so on. It also tells about features that must be included and features that could be included if time permits. Although it does not feel like something you can follow to get a finished product. Requirements is going to be established in iteration 2 but it feels like they should be included to be allowed to call it a project plan. The iterations feel more like a project plan, but they are under a separate headline.

2 | Iterations

This section holds the details of each iteration of the project.

2.1 Iteration 1

The task is to draft the documentations needed for the project, such as project plan and vision.

Resources used are lectures 1 and 2 and chapter 2,3,22 and 23of the course book. Reading of the book should take about 4 hours. The goal here is to do all documentations and start with some implementations before week 6.

Write vision: Project manager in association with other stakeholders should write a vision for the project. This should take about 30 minutes.

Write project plan: The project manager should then make a project plan. In this stage there should not be details about coding or modeling but more of a higher level of planning. For some parts of the project plan the manager must communicate with other resources and stakeholders.

- Write a justification why the game should be made – *After writing the vision the manager should have a view of why the game should be developed. This should take about 15 minutes.*
- List stakeholders – *Make a list of stakeholders in the project and specify which part of the project the different stakeholders should influence most. Also try to grade the different stakeholders influence. This should take about 30 minutes.*
- List resources and hard- and software requirements – *Which resources are available; how much time is there for the project? Which hard- and software requirements are there to develop the game? For this the manager will need help from the development team. This should take about 30 minutes.*
- List deadlines - *These are already decided by the customer, list them and specify shortly what is included in each deadline. This should take about 15 minutes.*
- Scope, Constraints and Assumptions – *List what must be included in the project and what should be included if possible. Also specify constraints that may affect the development of the project. To do this the manager will need help from the development team as well as the customer. This should take about 30 minutes.*

Write risk analysis: The manager then must make a risk analysis for the project.

- What are the risks?
- How to prevent risks?

- How to minimize the impact of risks?
- How to deal with risks?

To determine these different things the manager, again, will need help from the development team. This should take about one to two hours.

After completing all steps in iteration one, about two hours should be used to proofread the vision, project plan and risk analysis to make sure that everything is correct.

2.2 Iteration 2

The task is to create the different Use Case and start implementing them. Resourses are lectures in theme two and chapter 6, 7, 15 in the course book. Due to the volume of the book the chapters will be spread through iteration 2. Goal is to do all the implementation needed with the Use Case before week 9.

Basic:

- Write a fully dressed use case for a *basic* version of requirement Play Game. This should take 4 hours.
- Create a state machine for playing the game. This should take 2 hours.
- Implement the models. This should take 3 hours.
- Create a class diagram. This should take an hour.

Additional:

- Create extended use case diagram, this should take 30min.
- Write fully dressed use case for single player game, this should take an hour.
- Write fully dressed use case for multiplayer version. This should take an hour.
- Create state machine for single player version.
- Create state machine for multiplayer version. This should take an hour.
- Implement as much as possible of the extended models. This should take 4 hours.
- Create a class diagram. This should take an hour.

2.3 Iteration 3

The task is to test all the parts of the game and the features left from the previous previous iterations. Resourses are lectures in theme three and chapter 8 in the course book. Goal is to test everything and add some features before week 11.

- Write detailed planning for this iteration which should take about 30min.
- Design manual testcases for the use cases which was modeled in iteration two. This should take about 5 hours.
- Create automated unit-tests which also should take about 5 hours.
- Perform the manual testcases that was created and write down any problems. This should take about 30minutes.
- Inspect the code that is tested and look for improvements. This should take an hour.
- Write a test report after performing the tests that was designed in this iteration including any problems that came up. This should take about 30 minutes.

2.4 Iteration 4

The task is to complete anything left behind from the previous iterations. Goal is to have a complete and fully functioning Hangman game with its all features.

Handle comments about iteration 1-3

- Clarify main client and end users which should take 20 minutes.
- Update vision which should take 30min.
- Update project plan which should take 30min.
- Fix details in state machines, add pseudo states to make it easier to read. This should take 30min.
- Increase readability by adding more line-breaks and bold text to guide the tester which should take 20min.

Implement removing word from predefined list of word

• Write fully dressed use case for removing a word to the predefined list which should take 30min. estimated 2 hours each which result in 10 hours.

- Create state machine for removing a word to the predefined list which should take 30min.
- Implement removing a word from the predefined list which should take two hours.
- Update the class diagram which should take 15min.
- Design testcases for removing a word from the predefined list which should take an hour.
- Perform the manual testcases that was designed which should take 20min.
- Update the test report if any problems with the new implementation arise which should take 15min.

Implement possibility to play games until failure and earning a high-score

- Change the use case single player game to continue guessing words until player fails to guess a word and add a high score which is being enumerated for each win. This should take an hour.
- Update state chart for single player game which should take 30min.
- Implement the updated use case and state chart for single player game which should take 4 hours.
- Update test cases for use case single player game which should take an hour.
- Re-iterate testcases which should take 20min
- Update class diagram which should take 15min.

Finish project

- Add manual testcase for user case 6 Quit game which should take 30min.
- Clean up code and check methods documentation which should take 45min.
- Check project plan, diagrams and state charts which should take an hour.
- Check manual and automated tests which should take 30min.
- Perform all tests one final time which should take 30min.

- Write reflection with motivations for the diagrams, state machines tests and so on that was used during iteration 4 which should take 20min.

Reflections

I chose to implement two new features in this iteration since they seemed reasonably hard to implement. I decided to exclude for example username and high-score list since this seemed to complicate to implement and I prioritized planning and testing the features I do have instead of number of features.

In iteration 2 to have a unity in the project by using similar diagrams. The new features are similar in size and complexity to the previous features and needed about as much planning. Therefor I made use case diagrams and state machines. More diagrams would not result in a better result but mainly take time from my perspective. I also updated the class diagram.

I managed to implement a few more unit tests for the new features, mainly the remove word feature to see that the word really is removed. Otherwise I have mainly made manual test cases. I realize that this is not optional since manual test cases take a lot of time and is risky since tester might get sloppy, but since a lot of the game involve user input and prints to the screen I did not have the knowledge on how to automate this but still wanted coverage.

Reviewing the project and updating it makes me appreciate how much i have learnt. The game is finally complete, and I hope user will enjoy every bit of it because of its user friendliness.

3 | Risk Analysis

Risk is anything that has the tendency to influence the smooth flow of the project. It is important to be prepared to tackle them at any time this is termed as Risk management. Risk analysis is really difficult for me. To do the risk analysis, I must know the details of the rules, as well as and asses every step of the game in other to find the

risks. Since I just have skeletal design and code, I implemented the basic version as and when I know more, I will update it.

ID	Type of Risk	Probability	Impact	Strategies	How to Handle it
R1	Illness	4	5	Stay home and get drugs	Get permission from lecturer.
R2	Unsaved Changes	3	4	Save everything before taking breaks	Employ the use of recovery programs.
R3	Shortage of time	5	5	Plan ahead	Work during holidays and weekends.

Reflections

The hangman project is not that big, and it is obviously possible to do since it is an assignment, so it was a bit hard to come up with a large amount of risks. But it was still possible to come up with different strategies to prevent them and make the impact of them as little as possible. Many risks are things that feel kind of obvious and you usually just deal with them as they occur, so it was educational to write them down and analyze them.

4 | Time log

Iteration 1.

Task	Estimated time (hours)	Actual time Used (hours)	Reflections
Reading the course book	8	14	Since it was the first time, I spent more time than planned
Learn about rules of the game	2	1.5	
Plan for code	2	4	Took more time it just a rough

			estimation
Vision	1	1.15	
Risk Analysis	2	2.5	
Prof reading	1	1	

Iteration 2

Task	Estimated Time(hrs)	Actual Time(hrs)	Reflection
Code Writing	72	72	Had to go back after starting the multiple times so this actual time may not be exact.
Write a fully dressed use case for a *basic* version of requirement UC 2 Play Game	2	1.5	Fairly easy to draw
Create extended use case diagram	3	2	Fairly ok estimation. Changed this diagram several times.
Implement the models.	3	4	Time went faster than expected
Implement the extended models.	4	6	Not all extended model functions were implemented. Some functions was harder to implement than expected.

Create state machine(single player and multiplayer versions)	2	3	

It was very hard to measure time even though toggle was used. Especially when I was stressed and went back and forth between different tasks. The actual time may have been slightly underestimated but is as accurate as possible.

Iteration 3

Task	Estimated Time	Actual Time	Reflections
Writing Manual Test case	2hr	3hr	Some functions from the orginal UC was removed. But before this was done some time was consumed trying to solve such problems.
Unit TC	2hr	5hr	Time was consumed try to figure out what was testable because the implementation was not well done.
Running automated Test	1hr	10 min	Fairly easy.
Code Inspection	2hr	1hr	Parts of code had to be reconstructed to be testable.
Test Report	3hr	12hr	It took more time than planned because of attaching pictures to the report.

Iteration 4 Time Plan

Task	Estimated Time	Actual Time	Reflections
Write detailed	1	40	

planning for iteration 4			
Handled comments from iteration 1-3			
Clarify main client and users.	1	30 min	
Update vision	40min	30min	
Update project plan	30min	1	Completed the project.
Fixed details in Use case and state machine	1	40 min	
Increase readability in testing Implementing removing from the predefined list			
Updating Iteration 2			
Update the test report if any problems with the new implementation arise.			No problem to report.
Increased readability of iteration2	2	1	
Updating Iteration 3			
Perform the manual testcases that was designed.	30 min	40 min	-Waisted lots of time screenshotting and appending to the test document. - Much was not done here as all test were running as expected.
Rewriting testing code	1	1	
Perform automated test	20 min	10min	

Update the test report if any problems with the new implementation arise.	20 min	10 min	
Finished Project			
Clean code and methods- documentation.	1	45 min	
Check project plan, diagrams and state charts.	1	30	
Check manual and automated tests.	30	30	
Perform all tests one final time.	20	30	
Write reflection for iteration 4.	20 min	10min	