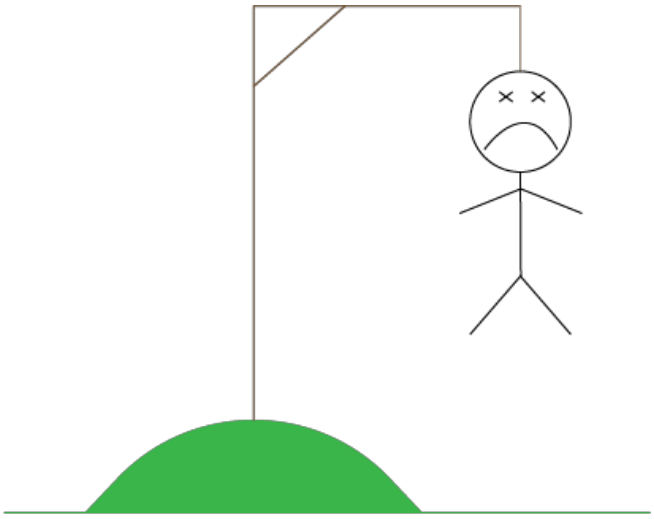
SOFTWARE DEVELOPMENT PROJECT HANGMAN

SIMON JONSSON

1DV600

190131



Contents

1 Revision History

Date	Version	Description	Author
22/2	1.1	Updated for iteration 2	Simon Jonsson
8/3	1.2	Updated Timelog	Simon Jonsson

2 General Information

Project Summary	
Project Name	Project ID
Hangman	1
Project Manager	Main Client
Simon Jonsson	Simon Jonsson
Key Stakeholders	
Simon Jonsson	
Executive Summary	

Project Hangman – v1.1 – Simon Jonsson – 190131

3 Vision

The project is to create a basic game of Hangman. Guess letters that the word contains and if the player guesses wrong a piece of the man to be hanged is displayed. If all pieces of the man is displayed, the player has lost.

The word to be figured out is to be randomly chosen.

The code is to be written in Javascript. The game should be written for the terminal and the man to be shown with images created with the keyboard for when the player guesses letters and answers wrong.

There should be functions covering for if the player guesses the same letter as before that stops the guess, letters already chosen should be displayed for the player.

There is to be a mandatory nickname for the player and also an option to be played with a friend.

My reflections

My reflection of the vision document is that it is great for the involved people to be on the same page regarding what is to be done, and what parts that should be included. The group then has a somewhat clear vision as to what is to be created and can continue work for the same goal.

If there is no clear vision to start with, the team can have different takes on what is to be done. Is it for the browser?, the terminal? Should there be images, what programming language to be used, what features the game should have and so o

4 Project Plan

The Project is to complete a game of Hangman to be run in the terminal. The project will be worked on by developer Simon Jonsson.

The project will be broken down to four iterations to complete the application in time for deadline. Start up iteration with deadline 8/2-19, make it ready for testing in the next iteration, deadline 22/2-19, testing and launching the final product, deadline 8/3-19.

The time estimated for the project is roughly 10 hours.

My reflections

A project plan is a great way to start off when trying to establish a contract with a customer. They get a clear view of how the work will be done and also an estimate for what the project will cost. I have not added any pricing in my plan, but an estimate of the time which is also good knowledge for the customer.

I believe the plan should include dates for when the iterations should be complete so that the customer is always up to speed on when to expect things to be done, which is great.

It is also good to follow up on your plan to change planning if necessary so the customer always have a clear part of the plan.

4.1 Introduction

A simple game of Hangman where you are to guess letters for a word.

4.2 Justification

As an assignment for the class 1DV600.

4.3 Stakeholders

Simon Jonsson - Developer

LNU - Project owner

4.4 Resources

Simon Jonsson - Developer.

4.5 Hard- and Software Requirements

Any modern computer with an IDE

4.6 Overall Project Schedule

8/2 - First review

22/2 - Second review

8/3 - Third and final review. Finished Product.

4.7 Scope, Constraints and Assumptions

The project consists of making an app where you play the game Hangman. It will be built with JavaScript and executed in the terminal. It will consist of one module and an API that supplies random words for the game. The game will be created for the course 1DV600 and will be completed on march 8th 2019.

To define something as out of scope it is a milestone that is not measurable.

---

5 Iterations

Plan for four iterations, including this. This is a fine-grained plan on what is to be done in each iteration and with what resources. To begin with, this is a plan of what we *expect* to do, update this part with *additions* (never remove anything) when plans do not match up with reality. Also make time estimates for the different parts.

In this course the overall planning has in some ways already been decided, so use the template to provide more details on specific tasks that define *your* project. Remember that you can plan to add features to any of the phases as long as the main focus is also met.

The first assignment is to complete iteration one.

5.1 Iteration 1

Complete documentation for a Project plan.

Start code with Javascript, set up a menu with options to set nickname, multiplayer, start game, quit game. Find an API to serve random words.

Add function to save nickname.

Shouldn't take more than two hours.

Resources - Simon Jonsson

5.2 Iteration 2

Add function to be able to start guessing letters and display your progress with the hanged man. Add function for going back to menu when the game is over. Add function for multiplayer option.

Time required, four hours.

Resources - Simon Jonsson

Diagrams Attached in git rep

5.3 Iteration 3

Testing, two hours.

Resources - Simon Jonsson

Try different choices in menu that aren't supposed to work. Guess letters that has been chosen, guess numeric values and symbols.

You may include additional features to the game in this iteration, but the main focus is on *testing*. Plan, perform and document your tests in this iteration.

5.4 Iteration 4

The outcome of this iteration is *the complete* game. Reiterate the steps in iteration 1 – 3 for a set of new features but also remember to see the project as a whole, not only its parts.

Resources - Simon Jonsson

---

6 Risk Analysis

My reflections

The use of a risk analysis is a great way to minimise risk from the start with strategies in place for any known risks that may occur. With a risk analysis we know the great risks from the start and can plan for them. We minimise uncertainty and potential loss. If we do not have a risk analysis in place it will most likely drain resources from the team and delay the project which could end with catastrophic results. With knowledge of risks comes higher value.

6.1 List of risks

Risk	Probability	Effects
Developer may fall out sick	Low	Tolerable
Developer may quit class	Insignificant	Catastrophic
Computer may break down	Low	Tolerable
Estimate of time needed	Low	Tolerable
Requirements change	Low	Tolerable

6.2 Strategies

Risk	Strategy
Developer may fall out sick	Developer must keep on working
Developer may quit class	Keep Developer happy and focused
Computer may break down	Have backup computer available
Estimate of time needed	Prepare with added time in planning
Requirements change	Prepare with added time in planning

7 Time log

	5/2		21/2	8/3				
	Estimated Hours	Actual Hours	Estimated Hours	Actual Hours	Estimated Hours	Actual Hours	Estimated Hours	Actual Hours
Set up menu	0,25	0,25						
Function nickname	0,25	0,25						
Function Start game	0,25	0,5						
Function quit game	0,25	0,5						
Function multiplayer			0,5	0,5				
Function play game			4	5				
Testing				6	2			
Complete Game							2	

8 Handing in

All assignments have a number of files to hand in. The overall advice is to *keep it simple*. Make it easy for the receiver to understand what the files are by using *descriptive* file names. Use as *few* separate documents as possible. Always provide a *context*, that is *do not* send a number of diagrams in "graphics format", but always in a document where you provide the purpose and meaning of the diagrams. Remember that the "receiver" is in reality a customer and as such has very little knowledge of the diagrams and documents – always provide context that make anything you hand in understandable to a non-technical person.

To hand in an assignment, make a git release and hand in the link via Moodle to that release.

Project Name – Version – Author – Date