1. Define development methodology:

Exactly like in a real company, you should adopt a certain development methodology such as (1) Waterfall or (2) Agile Scrum with its variant, (3) Test Driven Development (TDD).

1. produce a software requirements specification (SRS)
2. Write a test plan, including your test strategy and test cases.

you should also explain why you used a certain strategy or technique

it is wise to prioritize both your requirements and your test cases. Don’t forget to justify these decisions.

Evaluate your testing by using code coverage and mutation tools.

continuously monitor your testing process, for example by using a bug tracking system in order to extract metrics for your testing process.

Don’t forget to register the time you spent on each activity.

# Test plan Hangman1

**Introduction**

In this test plan the scope, approach, resources and schedule of Hangman1 are described. The application is a well-known game in which the computer chooses a word which is going to be guessed by the player. The computer informs the player about the length of the word beforehand. In each turn, the player guesses a letter, if the letter is in the word, the computer gives positive feedback, if the letter is not in the word, a limb of the player’s virtual body is hung. This goes on until the player has guessed the whole word or the player’s full virtual body is hanging.

The test items and test features are going to be: ….

Lower level test plans can be found in: ….

**Test items**

**Features to be tested**

The first feature that is going to be tested is whether the input of the user is correct. This should be tested in multiple ways, since the input can be invalid in multiple ways. The second feature that is tested is whether the game ends when it has to end and whether it gives the right output if it ends.

* Correct interface: Does the interface show the same number of underscores as the number of letters of the word? Does the right figure appear and do the limbs appear in the right order? Do the correctly guessed letters appear in the right way?
* Selection of words: Does the program randomly choose words for each game? Do the words have a minimal length of 2 letters?
* Valid input: does the input consist of 1 letter that is not in use already?
* Correct Feedback: when false letter is given, negative feedback is given. When right letter is given, positive feedback is given.
* End of game: does it end at right moment?

**Features not to be tested**

In this plan we will only test whether the program functions correctly. It will not be tested if the

* Whether the game starts on all systems: not in scope
* …

**Approach**

…

**Item Pass/Fail Criteria**

To pass a test case the

**Suspension Criteria and Resumption Requirements**

If one of the tests fail, suspend testing and fix the bug. When the developer thinks the bug is fixed, he/she should inspect the changed code to check whether no new bugs can have occurred from these changes. If the developer detects any possibilities for new bugs, he/she should test these changes in the code. When there are no possible new bugs from the changes that are made, the testing activities can be resumed.

**Test Deliverables**

The test deliverables consist of the following:

* This test plan
* Test design specifications
* Test case specifications
* Test procedure specifications
* Test item transmittal reports
* Test logs
* Test incident reports
* Test summary reports.

For each test case, the test input will consist of one word. The test output differs for each case. It can consist of the following **…..**

Test tools??

**Testing tasks**

To prepare for testing, one or more appropriate words have to be defined. In addition, the letters that the tester should insert have to be defined as well, since the letters that have to be inserted should differ for the word that is tested.

When the tester is going to test the program, it has to be opened on the computer that is going to be used and the tester needs a keyboard. The tester does not require any special skills.

**Environmental needs**

**Responsibilities**

**Staffing and Training Needs**

The skills the tester needs to have are that he/she can read and write in English and that he/she has minimal computer skills, i.e. can use the keyboard and applications on the computer. Therefore staffing needs by skill level or training options are not required.

**Schedule**

**Risks and Contingencies**

The assumptions made in this test plan is that the tester knows how to work with a computer, that the computer he/she tests on works and that he/she has reasonable knowledge of the game Hangman and the English language.

If the tester does not have these skills, there has to be found another tester. But since the testing requires basic knowledge and basic amenities, this should be easy.

**Approvals**

This plan must be approved by:

Date: Date: Date:

Name: Attila Schinagel Name: Dirk Zomerland Name: Nadine Enning

Date: Date:

Name: Roland … Name: Steven Raaijmakers.