

Section 7 - Testing

The WJEC specification says:

- a) Developmental testing: Produce evidence of testing at each stage of development.
- b) Testing the final system: Produce evidence of all problems encountered and actions taken to overcome these problems.
- c) Design a test plan to test:
 - - - each individual system function
 - that each individual function works with typical, extreme or invalid data
 - the whole system to ensure that the system produces the correct results for the data input.
- d) Actual test runs: Produce annotated test runs that include commentaries on the outcomes of the testing process.

Things to consider doing:

- i) Put a heading, **7a Testing at each stage.**
- ii) You need to provide evidence of testing at every stage. Use the test plans that you produce in Section 7c.
- iii) Put a heading, **7b Problems encountered and actions taken.**
- iv) When you do your testing, whatever test it is, you should take note of all problems encountered and what actions you took to sort out the problem. You need evidence. Did a test fail because you entered data incorrectly? Did a test fail because the code caused an error? What did you do about each problem.
- v) Put a heading, **7c Test plans.**
- vi) You should discuss your approach to testing. You may also want to describe why you will draw up your test strategy and plan before you begin coding. In your discussion here, you should mention the need to test as you develop code (white box testing) and give some examples of the kinds of things you would test and the kinds of tests you would carry out. You should also discuss the need to test the final solution against the Requirements Specification (black box testing) and how you would go about doing this. The third part of this sub-section is to describe user testing, why you would carry it out and how you will go about this. Your fourth area to consider is the need for acceptance testing; what it is and how you will carry it out. The test strategy is all about describing and justifying an overview to testing, a broad approach.

You should draw up three separate Test Plans, each done as a table.

Test Plan one will be Testing during development (white box testing). in a table, you should state what you will test, what data you will use, why you have chosen that data, what you expect the outcome to be and the results. There also needs to be a column for the page numbers of the evidence that the test has been carried out. You could plan to test some of your algorithms. You might want to fully test a couple of validation rules. Your teacher will advise you on the appropriate number of tests for your particular project. Make sure you include typical, extreme and invalid data for each data test.

Test Plan two will be Testing post development (black box testing). In a table, you should state each of your Requirement Specifications, the data you will use to test each one, why you choose that data, what you expect the outcome to be. There also needs to be a column for the page numbers of the evidence that the test has been carried out.

Test Plan three will be User and Acceptance testing. At the end of the project, you need to agree with the user that it does what you both agreed it would in the Requirements Specification. The user will therefore need to carry out the things identified in the Requirements Specification (even though you have done these tests yourself). If they are happy it all works, then they can sign it off. There also needs to be a column for the page numbers of the evidence that the test has been carried out.

vii) Put a heading, *7d Annotated test runs*.

viii) All your test runs from section 7c should be evidenced with annotated screendumps and cross-referenced to the relevant test. In your test plans, you should have said why you are doing each test. In your annotated evidence, you should describe the result and what it means. This is especially important for those test results that differ from the ones you predicted in your test plans.