

DECO3801/7381

TESTING PLAN & VALIDATION (10%)

Due 5pm, Tuesday 17 September 2013

Submit your functional and user test plans through blackboard.

Description

Testing your prototypes while in development is an essential safeguard for delivering a fully working, well-designed system. Testing will cover both functionality and user experience dimensions of your system, validated against your prototypes.

Remember that we do not care whether your prototype passes the tests! What is important is that your prototype is sufficiently **broad** to validate the plans you make for functional and user testing of the final product.

The aim of this testing plan is to:

- show us (and your client) how you can plan and conduct tests specifically relevant to your system and interpret test results to improve subsequent prototype versions
- provide an early check of the relevance, scope and reliability of your system

Deliverables

Your Test Plan assessment consists of:

1. Functional Testing Plan and Validation
2. User Experience Goals
3. User Testing Plan and Validation

Functional Testing Plan and Validation

This document covers the blow-by-blow testing of your system's functionality: Do your inputs and outputs act as expected and how well does your application handle any possible errors?

Begin with a high-level description of your unit testing strategy. You should describe your major application components (back-end code, javascript functionality etc.) and what testing frameworks and strategies you are implementing to test them.

Present a documented transcript of your test cases, in terms of which part of the system (e.g. module, method, combination of modules) they are testing, what they are testing for, and what the expected results are.

You should aim to deliver your functional testing as well-documented unit tests running against your prototype code base. You can choose to document your tests through comment strings within the test classes, but they should also be included in the test plan in tabular format. Your actual tests should be running against your prototype codebase.

It is recommended you build your unit tests alongside your codebase as you work. You will have more stable, happier code, and this component of the test plan will just involve ensuring your test methods are sensibly documented. It is also recommended that you

use code coverage tools where possible. They will help you discover aspects of your application that are under-tested and possibly unstable. Aim for high (90%+) test coverage.

There is no minimum or maximum number of tests required but you will be assessed based on how appropriate your test coverage is to your running prototype and to the project proposal. You can use code coverage to aid in demonstrating this.

Since your functional tests are (hopefully) running unit tests, your test transcript should be a simple matter of including the results of running your tests suite/s. However, you should document and discuss any unimplemented or failing tests in your test document. You should also discuss the implications that your functional tests have for the next version of your prototype.

User Experience Goals

This one-page document consists of a user experience profile for your product/system. What kind of experience is the use of your system intended to produce? What goals do you have for how people encounter and appreciate your interface? Your goals need to be product-specific, so generic goals like 'usable' and 'attractive' are not appropriate. Think in terms of the particular experiences a user will ideally encounter in use: engaging or surgical (in-and-out)? immersive or ambient? enthralling or passive? Think carefully about the impressions you want to create in use, ensuring they are appropriate to the functionality your system is offering.

This submission is a short, illustrated (with screenshots or storyboards) description of the experiences in use you are trying to generate with your interactive system and how they relate to the interactions you have designed into your interface.

User Testing Plan and Validation

Begin this document with a high-level summary of your user testing strategy, the prototypes you will test (using screenshots where appropriate), and the coverage of those tests. NOTE: Each group must enlist 4-6 users *per user group*. Your users must be recruited from outside the DECO3801 cohort. You will be rewarded for involving potential users and conducting tests in legitimate contexts of use.

Describe a minimum of five scenarios of use, consisting of:

- 3-4 typical use scenarios—bread-and-butter sequences of use through your system, of how people encounter the central functions of the system through the tasks that they will perform with it.
- 1-2 first encounter scenarios—take us through how someone is first introduced to the system, installs and opens it for the first time, and learns through use what it is for and how it works.
- 1-2 fringe or limit cases—identify possible scenarios of use or mis-use that push the limits of your system. These should not be about technical issues (network failure etc.), but about use cases that arise in the ordinary circumstances that pose issues for the idealized use of the functionality you have implemented.

Scenarios can be short textual narratives or visual storyboards or both. Each scenario should specify its particular user (e.g. a brief persona) and context of use.

Describe your scenario-based user test (as we discussed in class), justifying how you will be carrying out your tests and what specific UX aspects of your system you will be

investigating. Explain the environment for testing, and your methods of observation, evaluation and documentation.

Identify the qualitative and quantitative metrics you will use to evaluate the prototype in these tests.

Summarise the results of your tests. For each scenario-based test, tabulate your metric results and qualitative observations against the relevant task or system element in the scenario. Present your users' feedback from the tests. Close with a general discussion that identifies the range of issues raised from the tests, identifies areas for improvement, and describes design recommendations that arose from evaluating the prototype with users. Outline any modifications you wish to make to the scenarios or tests when conducting them with your final prototype.

Marking Rubric (out of 100)

Criteria	Poor	Satisfactory	Good
Functional Testing – out of 50			
Testing coverage	Important parts of solution not covered by functional tests.	Good, but incomplete, coverage of solution by functional tests.	Thorough coverage of solution by test plan, including difficult to foresee problems.
Testing methods	Significant problems with approach to functional test execution.	Solid approach to functional test execution.	Thorough consideration and application of functional testing methods.
Testing results	Little learned from functional tests, or few/no implications developed from test results. Prototype not appropriately used to validate test plans.	Appropriate implications for future testing and system design derived from test results and analysis. Satisfactory use of prototype in tests.	Insightful consequences for future functional testing, system design and functionality from test results. Well considered use of prototype.
User Experience Goals – out of 10			
UX goals	Poor match between user experience goals and functionality. Description of interface unclear.	Good match between UX goals and proposed system, illustrated and described clearly.	Original user experience goals finely tuned to the system being designed. Well described and clearly illustrated.
User Testing – out of 40			
Scenarios and Testing coverage	Significant usage scenarios not generated and/or not covered by user tests.	Sound coverage of usage in scenarios generated and tested with users.	Thorough coverage of usage represented in the scenarios generated and tested with users, including less obvious usage.
Testing methods	Significant issues or shortcomings with approach to carrying out user testing.	Appropriate approach to conducting user tests, including actual users involved and contexts of use for tests.	Detailed and considered approach to selecting users, testing in use contexts and carrying out user tests.
Testing results	Little learned from user tests, or little appreciation for how results affect proposed system under development. Prototype or prototype used in tests not appropriate.	Appropriate implications for future testing and system design derived from test results and analysis.	Insightful consequences for future user testing, system design and functionality from test results.