Milestone 6
Software Test Plan
CSCI 3130 – Summer 2011
Due: July 15, 2011 (3pm), 10% of term mark

Purpose: To develop a software test plan.

Resources : We have discussed elements of a software test plan throughout the semester and much of this deliverable could have been written when the SRS and SDS were completed. The testing slide deck has further details as well as a link to a template. The handouts from the class on June 28th also provide background about the types of testing and their roles, as well as various techniques for ensuring comprehensive testing and a running example showing how tests are related to requirements and design.

There are many templates available for you to choose between, or you can come up with your own design:

- http://www.sqatester.com/index.htm This site has many resources and test examples (e.g., test matrix, test cases), as well as test plans , including the IEEE spec: http://www.sqatester.com/documentation/downloads/IEEEStandardTestPlans.doc

- http://www.construx.com/Page.aspx?nid=208 This site has several templates and checklists that you may find useful. The Test Plan doc has a nice set of instructions built into it to help you fill in the sections.

Deliverables:

Post your test plan on your website and send the link to it to Dinesh. Drop a hard copy of the test plan in the dropbox by 3pm.

At a minimum, **for each type of testing** (**unit test, integration test, system test** – your demo will be the acceptance test), I would like to see the following information:

1) what tests you plan to run
2) the order to run them in
3) the test materials needed for each test (e.g., populated data bases, scripts, and how long they will take to prepare)
4) cross reference requirements (from SRS) and design elements (from SDS) against tests as appropriate (e.g., a matrix – requirement 1.1 tested by test script 3)
5) predict expected results for each test and how long that test will take to run
6) At least 1 of the unit tests should be fully specified with both black box and white box test cases. Enough analysis should be provided (in an Appendix) so that it is clear how these test cases provide coverage (whitebox: statement, branch, path; blackbox: equivalence partitioning/boundary values) (see handout for a running example). You may be more terse on

others and it is appropriate to have some without a suite a test cases yet (but there should be a plan in place for their generation and they should be specified in version submitted in the final report).

Marking:

As usual, each work product will be marked on the following basis:

- Completeness
- Consistency
- Level of specificity
- Clarity
- Organization
- Format, spelling, grammar
- Use of diagrams (if appropriate)
- Software craft

Aspects of software craft that may be pertinent to this milestone, include:

- Use of a version control system
- Use of a build system
- Use of a testing framework (Unit, Regression, UI monkey…)
- Use of a ticket tracking and release system for defects

As your group members inspect the test plan before submitting it, think about whether you have done a thorough enough job that if your TA was hired to test your software, he would be able to test it sufficiently by following your plan and implementing your test cases.