Testing

This will be posted to Piazza

The Internal Test Plan template will be posted to Piazza

Preliminary internal test plan is due start of class Wed. 2/25

Why Test?

- Uncover bugs either in a module or how the modules are integrated

- Cannot test in quality!!

Principles (Pressman, 4th Edition)

- All tests should be traceable to the customer requirements
- Tests should be planned for long before testing begins
- The Pareto principle applies to software testing
- Testing should be “in the small” and progress towards testing “in the large”
- Exhaustive testing is not possible
- To be most effective, testing should be conducted by an independent third party

Testing

- Software engineering test methods from class tend to focus on **operability** (correctness of the internal software)
  - Unit progressing to integrative testing
  - **I assume you will do this through the actual programming process**
- Systems and product development focus on **observability** (correctness of what the stakeholder sees and uses)
  - More detailed than the the final user acceptance test

Verification versus Validation

- **Verification** is that the system correctly meets all the requirements
  - Are we building the product right?
- **Validation** is that the system meets the intended use
  - Are we building the right product?
- Do not confuse V&V with operability and observability, both are verification that the product meets the agreed upon requirements

Examples of Metrics

- Score
- Time to completion
- Number of clicks
- Number of program failures
- Mean time between failures
- Number of dropped connections

Quantitative, objective, measurable, relevant

Test Methods (aka experiments)

- Repeatable and allows for regression
- Should contain
  - Step-by-step procedure
    - Input, output procedure
    - This could be generated with a black-box, white-box, or other analysis
    - Can be automated
      - Hmmm, should you have two test sets? three?
  - Equipment, setting
  - What you are testing or expect to change (dependent variables)
    - Ex.
  - What you are holding fixed (independent variables)
    - Ex. Same OS

Example

Test 2 Generation of Flow Rate Graph

- Input: a test set of 20 legal inputs, 10 illegal inputs. The 20 legal inputs will have 16 chosen at random from the algorithm's boundary conditions and 4 chosen at random from nominal or steady state conditions.
- Procedure: The main program with all other functions as stubs will read each input from the test set file and write to an output file. The main program will run on Windows 8, VSB version N.
- Output: The result of the test will be a) no execution error, correct value or action; b) no execution error, incorrect value or action; or b) execution error. The program must produce a) for all members of the test set. If the program does not pass, the same test set will be used in subsequent tests.

Common Mistakes in Testing

- Underestimating the amount of time to execute a test and thus promising more tests or test input than is realistic
- Not including illegal/accidental and boundary/ edge conditions
- Not following the same procedure and test sets (no regression testing)

Template Overview

1. State requirements using the same numbers from your requirements document (i.e., cut and paste)
2. State the prototype and how it will meet the key requirements.
3. Test based on observability
   - What key requirements will be tested by this test: *3.3.1-3.3.4, 3.4.4*
   - Metric: *Correctly computed graph values*
   - Test method and number of times it will be repeated
   - Who will be the independent third party conducting the test; if no third party, provide justification