CISC327 - Software Quality Assurance

Lecture 9

Introduction to Systematic Testing, part 2

Introduction to Systematic Testing

- Outline
  - Last time we began with basic definitions, validation & verification, the role of specifications, and the levels of testing
  - Today we continue with:
    - Testing in the life cycle
    - Test design and strategy
    - Test plans and procedures
    - Test results

What is Systematic Testing?

- An explicit discipline or procedure (a system) for
  - choosing and creating test cases
  - executing the tests and documenting the results
  - evaluating the results, possibly automatically
  - deciding when we are done (enough testing)

Validation vs. Verification

- Verification
  - Check that the software meets its stated functional and non-functional requirements
- Validation
  - More general than verification, ensure that the software meets the customer's expectations
  - Requirements specifications do not always reflect the real wishes or needs of system customers and users

Testing in the Life Cycle

- Kinds of Tests
  - Testing has a role at every stage of the software life cycle
  - As we have seen, tests play a role in the development of code (unit testing), the integration of the units into subsystems (integration testing), and the acceptance of the first version of the software system (system testing)
  - **Black box** testing methods are based on the software's specifications
  - **White box** (or glass box) testing methods are based on the software's code

-ility Testing

- Identify the -ilities
  - General way of specifying system characteristics for quality or testing
    - Capability: Can the system perform the required functions?
    - Reliability: Will it work well and resist failure in all required situations?
    - Usability: How easy is it for a real user to use the product?
    - Performance: How fast and responsive is the system?
    - Security: Is the system appropriately secure?

Testing in the Life Cycle

- Regression Tests
  - In addition, as the system is maintained, other kinds of tests based on past behaviour come into play
  - Once a system is stable and in production, we build and maintain a set of regression tests to ensure that when a change is made, the existing behaviour has not been broken
  - These often consist of a set of actual observed production inputs and their archived outputs from past versions of the system

Testing in the Life Cycle

- Failure Tests
  - As failures are discovered and fixed, we also maintain a set of failure tests to make sure that we have really fixed the observed failures and to make sure that we don't cause them again
  - These consist of a set of actual observed inputs that caused the past failures and their archived outputs after the system was fixed

Test Design

- Design of Tests
  - The design of tests for a system is a difficult and tricky engineering problem,
    as important as design of the software itself
  - The design of effective tests requires a set of stages from an initial high level test strategy down to detailed test procedures
  - Typical test design stages are:
    - test strategy
    - test planning
    - test case design
    - test procedure

Test Strategy

- Test Strategy
  - A test strategy is a statement of the overall approach to testing for a software development organization
  - Specifies the levels of testing to be done as well as the methods, techniques, and tools to be used
  - Part of the project's overall quality plan, to be followed and reported by all members of the project

Test Plans

- Test Plans
  - A test plan for a development project specifies in detail how the test strategy will be carried out for the project
  - In particular, it specifies:
    - the items to be tested
    - the level they will be tested at
    - the order they will be tested in
    - the test environment
  - May be project-wide, or may be structured into separate plans for unit, integration, system, and acceptance testing

Test Case Design

- Test Case Design
  - Once we have a plan, we need to specify a set of test cases for each item to be tested at each level
  - Each test case specifies how the implementation of a particular functional requirement or design unit is to be tested and how we will know if the test is successful
  - Usually a single step or small sequence of steps to determine if a feature of an application is working correctly

Test Case Design

- What might a test case look like?

  Test: login command, agent

  Purpose: check that login is accepted

  Input:

  login

  agent

  logout

  Input files: Valid Services File with no accounts in it

  Output files: Transaction Summary File with no transactions

  Output: none; possibly information messages in response to commands

Test Case Design

- Test Case Design (continued)
  - It is important to include test cases to test both that the software does what it should (positive testing, like the previous slide) and that it doesn't do what it shouldn't (negative testing)
  - Test cases are specified separately at each level: unit, integration, system, and acceptance
  - The test case documentation for each level form a test specification for the level

Test Procedures

- Test Procedures
  - The final stage of test design is the test procedure, which specifies the process for conducting test cases
  - For each item or set of items to be tested at each level of testing, the test procedure specifies the process to be followed in running and evaluating the test cases for the item
  - Often this includes the use of test harnesses (programs written solely to exercise the software or parts of it on the test cases), test scripts (automated procedures for running sets of test cases), or commercial testing tools

Test Reports

- Documenting Test Results
  - Output of test execution should be saved in a test results file, and summarized in a readable report
  - Test reports should be designed to be concise, easy to read, and to clearly point out failures or unexpectedly changed results
  - Test result files should be saved in a standardized form for easy comparison with future test executions

Summary

- Introduction to Testing
  - Testing is not just a one time task, it is a continuous process that lasts throughout the software life cycle
  - Effective testing requires careful engineering, similar and parallel to the process for design and implementation of the software itself
  - An overall test strategy drives test plans, test case design, and test procedures for a project

Summary

- References
  - Sommerville, ch. 8, "Software Testing"
  - The Software Test Page (on the web)
- Next
  - Introduction to Black Box Testing
  - Assignment #1 due **next** Thursday