SE 2AA4, CS 2ME3 (Introduction to Software Development)

Winter 2018

31 Overview of Testing (Ch. 6)

Dr. Spencer Smith

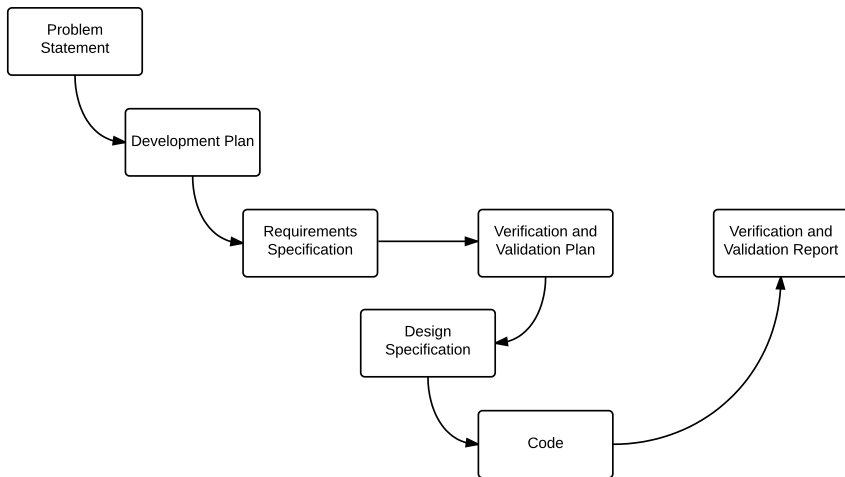Faculty of Engineering, McMaster University

March 26, 2018

31 Overview of Testing (Ch. 6)

- Administrative details
- Rational design process review
- Types of test
  - White box versus black box
  - Manual versus automated
  - Static versus dynamic
  - Continuous integration testing
- Fault testing
- Homework problem

Administrative Details

- Today's slide are partially based on slides by Dr. Wassyng
- A2: Solution pushed to repo
- A3: Part 2 - Code: due 11:59 pm Mar 26
- A4: Due April 9 at 11:59 pm
- No classes on Thurs, Mar 29 or Fri, Mar 30
- Final tutorial (examination review)
  - ▶ Monday, Mar 26
  - ▶ Tuesday, Mar 27
  - ▶ Friday, Apr 6 (no tutorial on Fri, Mar 30)
- Course Evaluation
  - ▶ https://evals.mcmaster.ca
  - ▶ Start: Tues, Mar 27, 10:00 am
  - ▶ Close: Tues, Apr 10, 11:59 pm
  - ▶ *Grade bonus for class participation!*

Rational Design Process

White Box Versus Black Box Testing

- Do you know (or can you guess) the difference between white box and black box testing?
- What if they were labelled transparent box and opaque box testing, respectively?

White Box Versus Black Box Testing

- White box testing is derived from the program's internal structure
- Black box testing is derived from a description of the program's function
- Should perform both white box and black box testing
- Black box testing
  - ▸ Uncovers errors that occur in implementing requirements or design specifications
  - ▸ Not concerned with how processing occurs, but with the results
  - ▸ Focuses on functional requirements for the system
  - ▸ Focuses on normal behaviour of the system

White Box Testing

- Uncovers errors that occur during implementation of the program
- Concerned with how processing occurs
- Evaluates whether the structure is sound
- Focuses on abnormal or extreme behaviour of the system

Dynamic Testing

- Is there a dynamic testing technique that can guarantee correctness?

- If so, what is the technique?

- Is this technique practical?

Dynamic Versus Static Testing

- Another classification of verification techniques, as previously discussed
- Use a combination of dynamic and static testing
- Dynamic analysis
  - Requires the program to be executed
  - Test cases are run and results are checked against expected behaviour
  - Exhaustive testing is the only dynamic technique that guarantees program validity
  - Exhaustive testing is usually impractical or impossible
  - Reduce number of test cases by finding criteria for choosing representative test cases

Static Testing Continued

- Static analysis
  - Does not involve program execution
  - Testing techniques simulate the dynamic environment
  - Includes syntax checking
  - Generally static testing is used in the requirements and design stage, where there is no code to execute
  - Document and code walkthroughs
  - Document and code inspections

Manual Versus Automated Testing

- What is the difference between manual and automated testing?
- What are the advantages of automated testing?
- What is regression testing?

Manual Versus Automated Testing

- Manual testing
  - Has to be conducted by people
  - Includes by-hand test cases, structured walkthroughs, code inspections
- Automated testing
  - The more automated the development process, the easier to automate testing
  - Less reliance on people
  - Necessary for regression testing
  - Test tools can assist, such as Junit, Cppunit, CuTest etc.
  - Can be challenging to automate GUI tests
  - Test suite for Maple has 2 000 000 test cases, run on 14 platforms, every night, automated reporting

Automated Testing at MapleSoft

- Three steps
  - Write the problem description
  - result := solver(problem)
  - assert(result == expected)
- Assert writes out code to reproduce any failures
- Track failures
  - Source code management (like subversion or git)
  - Database of test cases, functions called
  - Database of source files, functions defined
  - Database of 40 days of timings and resources used
- Automatically sends an e-mail to the programmer and his/her boss

Automation Cases

- How would you automate an application that does image processing (filtering, edge detection, format conversion etc.)?

- Can you automate testing the game play for a game?

Continuous Integration Testing

- What is continuous integration testing?

Continuous Integration Testing

- Information available on Wikipedia
- Developers integrate their code into a shared repo frequently (multiple times a day)
- Each integration is automatically accompanied by regression tests and other build tasks
- Build server
  - Regression tests
    - Unit tests
    - Integration tests
    - Portability tests
  - Static analysis
  - Profile performance
  - Extract documentation
  - Update project web-page
  - etc.
- Addresses sync between baseline and a developer's code

Continuous Integration Tools

- Gitlab
  - Example at Rogue Reborn
- Jenkins
- Travis
- Docker
  - Eliminates the "it works on my machine" problem
  - Package dependencies with your apps
  - A container for lightweight virtualization
  - Not a full VM

Quality Testing: Installability

- How might you test installability?

Fault Testing

- Common analogy involves planting fish in a lake to estimate the fish population
- T = total number of fish in the lake (to be estimated)
- N = fish stocked (marked) in the lake
- M = total number of fish caught in lake
- M' = number of marked fish caught
- $T = (M - M')*N/M'$
- Artificially seed faults, discover both seeded and new faults, estimate the total number of faults

Fault Testing Continued

- Method assumes that the real and seeded faults have the same distribution
- Hard to seed faults
  - By hand (not a great idea)
  - Independent testing by two groups and obtain the faults from one group for use by the other
- Want most of the discovered faults to be seeded faults
- If many faults are found, this is a bad thing
- The probability of errors is proportional to the number of errors already found

Exam Question: Homework

After completion of a complex software project, two independent groups test it. The first group finds 300 errors and the second group finds 200 errors. A comparison of the errors discovered by the two teams shows that they found the same error in 50 cases. What is the range that estimates the number of errors in the original software project?

A. 900–1000

B. -17–25

C. 600–1500

D. 0–150

E. 150–250