Software Verification & Validation

Natthapong Jungteerapanich

Handout 9

Regression Testing

Acknowledgement

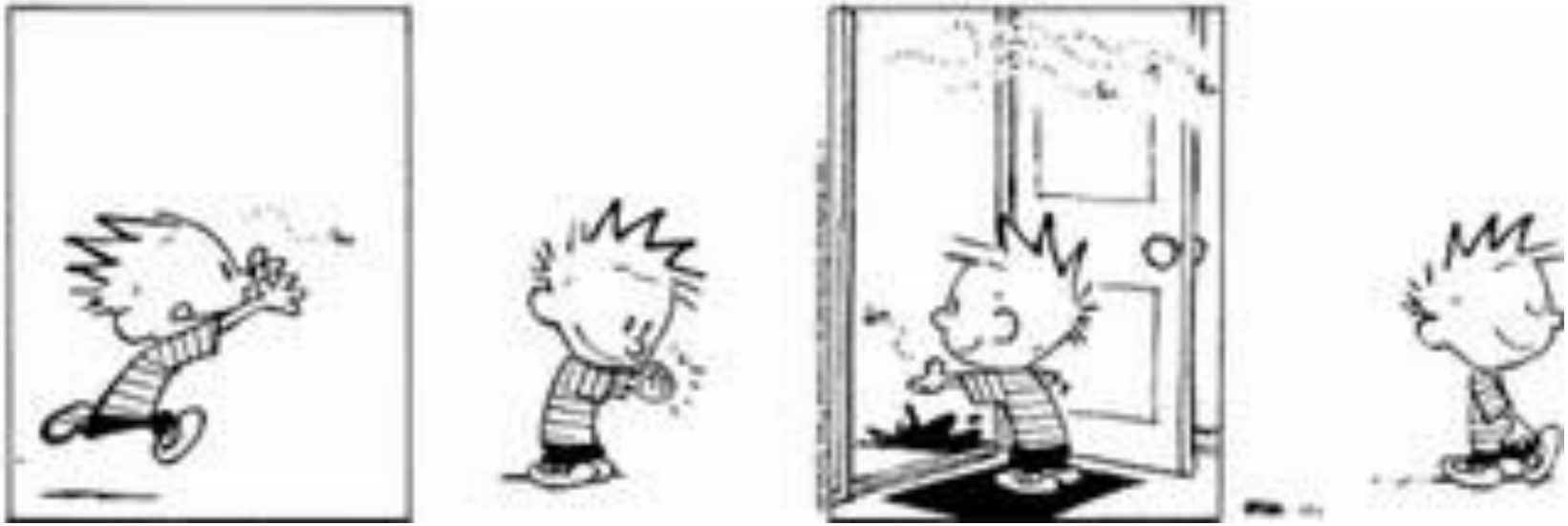
- Some slides in this lecture are adapted from

  - **Paul Ammann and Jeff Offutt**'s slides for their textbook **"Introduction to Software Testing"**

  - **Prof. Lori A. Clarke**'s slides for her course **"CS521-621 : Advanced Software Engineering: Analysis and Evaluation"** at University of Massachusetts Amherst

Regression Testing

Regression:
"when you fix one bug, you introduce several newer bugs."



* Taken from http://got-bugs.blogspot.com/2009/09/example-of-why-regression-testing-is-so.html

The Bug Song

- 99 little bugs in the code,
- 99 bugs in the code,
- Fix one bug, compile it again,
- 101 little bugs in the code.
- 101 little bugs in the code,
- 101 bugs in the code,
- Fix one bug, compile it again,
- 103 little bugs in the code.
- ...

*** Taken from http://www.tohlejokes.com/**

Regression Testing

Regression Testing is the process of re-testing software that has been modified

- Most software today has very little new development
  - Correcting, perfecting, adapting, or preventing problems with existing software
  - Composing new programs from existing components
  - Applying existing software to new situations
- Because of the deep interconnections among software components, changes in one method can cause problems in methods that seem to be unrelated
- Not surprisingly, most of our testing effort is regression testing
- Large regression test suites accumulate as programs (and software components) age

Regression Testing

- Regression testing primarily involves
  - Selecting from existing test cases,
  - Adding some new test cases,
  - Deleting or updating some old test cases.

- The purpose of performing regression testing is to instill confidence that changes to the code are correct, i.e.
  - new functionalities and corrected/modified functionalities behave as they should
  - unchanged functionalities are indeed unchanged

Regression Test Selection Techniques

Change Impact Analysis

How does a change impact the rest of the software ?

- When a small change is made in the software, what portions of the software can be impacted by that change ?
- More directly, which tests need to be re-run ?
  - Conservative approach : Run all existing tests
  - Random approach : Randomly select some portion of existing test cases to run
  - Is there a more effective and realistic approach to select test cases?
- Clearly, tests that never reach the modified statements do not need to be run

Test Selection Criteria

- Suppose P is the code before changes were made which has been tested with and passed a set T of test cases.

- Suppose P' is the code after changes were made. Which test cases in T should be executed during regression testing?

- A test case t ∈ T is modification-revealing if it produces different outputs for P than for P'
  - A modification-revealing test case reveals the change in the output as a result of changes in the code.
  - In general, we cannot determine which elements of T are modification revealing
- A test case t ∈ T is modification-traversing if it executes a statement in P' that has changed
  - Modification traversing approximates modification revealing. A test case which traverses a modified part of the program may not reveal any change in the output.
  - Can be computed

Test Selection Criteria

What's to consider when selecting existing test cases to retest the revised program with?

- Inclusive : A selection technique is *inclusive* if it includes all the test cases that are modification revealing.

- Precise : A selection technique is *precise* if it omits all test cases that are *not* modification revealing.

- Efficient : A selection technique is *efficient* if deciding what tests to omit is cheaper than running the omitted tests
  - This can depend on how much automation is available

Managing Tests in a Regression Test Suite

- Test suites accumulate new tests over time
- Test suites are usually run in a fixed, short, period of time
  - Often overnight, sometimes more frequently, sometimes less
- At some point, the number of tests can become unmanageable
  - We cannot finish running the tests in the time allotted
- We can always add more computer hardware
- But is it worth it?
- How many of these tests really need to be run?

Policies for Updating Test Suites

- Which tests to keep can be based on several policies
  - Add a new test for every problem report
  - Ensure that a coverage criterion is always satisfied

- Sometimes harder to choose tests to remove
  - Remove tests that do not contribute to satisfying coverage
  - Remove tests that have never found a fault
  - Remove tests that have found the same fault as other tests

- Test Case Prioritization
  - Assign priority level to each test case based on its effectiveness history.
  - Execute test cases with higher priority levels first.

Automation and Tool Support

Regression tests should be automated

- Too many tests to be run by hand.
- Tests must be run and evaluated quickly
  - often overnight, or more frequently for web applications
- Testers do not have time to view the results by inspection
- Types of tools :
  - Test environment or infrastructure support
    - Capture / Replay – *Capture* values entered into a GUI and *replay* those values on new versions
    - Scripting software – Manages the process of obtaining test *inputs*, *executing* the software, obtaining the *outputs*, *comparing* the results, and generating *test reports*
  - Test data selection support
    - Coverage Record – Record the code coverage information of each test, which is later used to determine which test to run during regression testing
    - Version control – Keeps track of collections of *tests*, expected *results*, where the tests *came from*, the *criterion* used, and their past *effectiveness*

Summary of Regression Testing

- We spend far more time on regression testing than on testing new software
- If tests are based on covering criteria, all problems are much simpler
  - We know why each test was created
  - We can make rationale decisions about whether to run each test
  - We know when to delete the test
  - We know when to modify the test
- Automating regression testing will save much more than it will cost