Topics for this Lecture

- What is regression testing?
- When to do it?
- What is the strategy?

Oregon State University

Regression testing

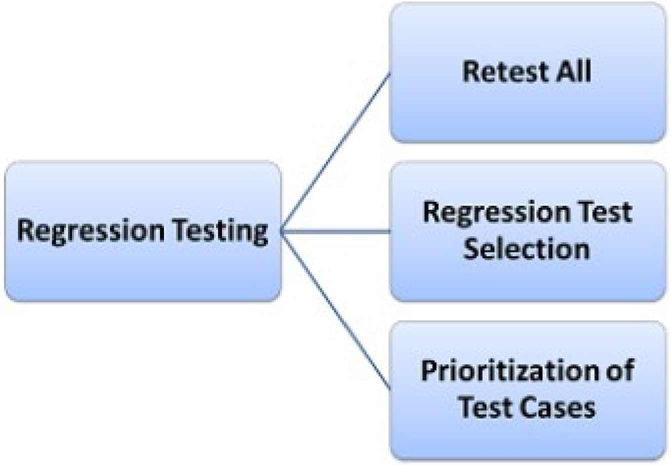
- A program **P** has been modified into program **P`** (maintenance)
- **P`** needs specific testing attention to ensure that
  1. newly added or modified code behaves correctly
  2. code carried over unchanged, continues to behave correctly
- **Regression testing** is a type of software testing which verifies that software which was previously developed and tested still performs the same way after it was changed.

When to do it

- Regression Testing is required when there is
  - Changing, adapting software to new conditions
  - Adding new feature to the software
  - Fixing other bugs
  - Performance issue fix
- Basic Problems of Regression Test
  - If I change feature X, how many test cases must be revised because they use feature X?
  - Which test cases should be removed or replaced?
  - Which test cases should be added?

What’s the strategy

- **Retest All**
  - This is one of the methods for regression testing in which all the tests in the existing test suite should be re-executed. This is very expensive as it requires huge time and resources.



Regression Test Selection: Instead of re-executing the entire test suite, it is better to pick and run test cases that test new changes functionality/code.

- **Safe:** selects all the test cases that cover/execute the changed methods at least once.
- **Minimization:** selects a minimum set of test cases that execute all the changed methods.

Regression Test Prioritization Technique

- **Test prioritization technique** ranks test cases for execution in order to attempts to increase their effectiveness in meeting some performance goal.

- Test prioritization requires criteria for ranking
  - **Criteria 1**: cost (e.g., execution time)
    - Tests with lower costs are ranked first while test with higher costs are ranked last.
  - **Criteria 2**: risk (expected risk of not executing a test)
    - Tests that may find failures with a high risk get higher priority than test that may find failures with low risks.

References:

Code Complete: A Practical Handbook of Software Construction, Second Edition 2nd Edition by Steve McConnell
Ammann, Paul, and Jeff Offutt. Introduction to software testing. Cambridge University Press, 2008.
Patton, R. (2005). Software Testing (2nd ed.).