Defining Regression Testing-I

• Regression testing is the execution of an ensemble of test cases on a program in order to ensure that its revision does not produce unintended faults, does not "regress" - that is, become less effective than it has been in the past.

Acceptable Costs of Selecting/Prioritizing & Executing Test Cases

Acceptable Fault Detection Ability

Constraints

Cost

Effectiveness

Values
• Problem (Rothermel et al):
  Given program P, its modified version P’, and a test set T that was used to previously test P, find a way to utilize T to gain sufficient confidence in the correctness of P’
Regression Testing Process

- Identify and modify/remove the obsolete test cases from \( T \) if specifications have changed.
  - Test case revalidation problem

- Select \( T' \subseteq T \), a set of test cases to execute on \( P' \)
  - Regression test selection problem
  - Test \( P' \) with \( T' \), establishing \( P'' \)'s correctness w.r.t. \( T' \)
    - Test suite execution problem

- If necessary, create \( T'' \), a set of new functional or structural test cases for \( P' \)
  - Coverage identification problem
  - Test \( P' \) with \( T'' \), establishing \( P'' \)'s correctness w.r.t. \( T'' \)
    - Test suite execution problem

- Create \( T''' \), a new test suite and test execution profile for \( P' \), from \( T, T', \) and \( T'' \).
  - Test suite maintenance problem
Regression Testing Techniques

- Regression test selection techniques- “Screen”
- Test case prioritization techniques- “Order”
  - General vs. version-specific
- Test suite reduction techniques- “Remove”
  - Reduce testing costs by permanently eliminating redundant test cases from test suites in terms of code or functionalities exercised.
Regression Test Selection
(Rothermel et al)

• Select $T'$ to be a subset of $T$ and use this suite for regression testing purposes
• Cost-effective if selection process does not outweigh execution of additional tests
• Many different techniques that generally reply upon the source code of software
Regression Test Selection (RTS) Techniques

• Minimization techniques
  • Coverage of modified or affected portions of P (at least once).

• Dataflow techniques
  • Exercising of data interactions (def-use) being affected by modifications

• Safe techniques
  • The selected subset T’ contains all test cases in T that an reveal faults in P’
  • The selection of every existing test case that exercises any program entity that could be affected by a given change.
    • Control-flow/statement-based Dejavu (Harrold et al)
    • Functions/global vars/macros/type definitions-based TestTube (Chen, Rosenblum & Vo)

• Ad hoc/Random techniques

• Retest-All technique
How does Safe RTS work?

- Assumption 1: P-Correct-for-T
- Assumption 2: Obsolete-Test-Identification

- Assumption 3: Controlled RT
  - Operating env
  - Non-deterministic statement ordering
  - DB & files that contribute data
  - .....

Figure 1: Relationship between classes of tests.

Regression Test Prioritization (Elbaum et al)

- Order the execution of a regression test suite so that the “best” tests run first
- Prioritize based on code coverage, fault-exposing potential, etc.
- Hopefully, defects are revealed earlier in the regression testing process
Selection and Prioritization

• These two techniques can be used in isolation or in conjunction
• This represents an advanced, but possibly incomplete, solution to regression testing problem
Cost-Effectiveness Prediction

- A predictor is inexpensive to apply but indicates whether using a selective regression testing method is likely to be worthwhile.
  - Average coverage-based predictors (Rosenblum & Weyuker)
  - Incorporating distribution of modification (Harrold et al.)

<table>
<thead>
<tr>
<th>Selecting/Prioritizing test cases</th>
<th>Running selected test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(n’ faults found)</td>
</tr>
</tbody>
</table>

Selecting/Prioritizing test cases | Running selected test cases |
-----------------------------------|-----------------------------|
                                   | (n’ faults found)           |

Running all test cases (n faults found)

0 hour                               | Cost/time                  | k hours
Empirical Studies

- When regression faults are revealed by certain test cases, path spectrum differences are likely also to be revealed by them, though the reverse is not true. (Harrold et al.)
- The impacts of test suite granularity on the cost-effectiveness of regression testing (Rothermel et al.)
- The effects of test application frequency/amount of change made between RTs on the costs and benefits of RTS techniques (Kim, Porter, & Rothermel)
- …..
Current Regression Testing Challenges

(Kapfhammer)

- Safe regression testing is code based and requires a controlled environment.
- Simple change of OS, virtual machines, device drivers, or file systems often nullify selection potential.
- Industry experience and intuition indicate that software environment matters.
- Regression testing prioritization is promising, yet very computationally intensive (mutation analysis).