Regression Test Selection of Manual System Tests in Practice

Elmar Juergens, Benjamin Hummel
Florian Deissenboeck, Martin Feilkas
Christian Schlögel, Andreas Wübbeke
Regression Test Selection @ Wincor Nixdorf

System Tests
• Performed manually (machine interaction)
• All tests executed for each major release
• Take several person months to execute

Hotfix Scenario
• Rapidly deploy fix for problem in the field
• Fixing a known bug can break existing code
• Unfeasible to perform all tests

Which tests are most likely to uncover errors introduced during fixing?
Selective Regression Testing (SRT): Overview

Safe SRT → error-uncovering test cases ⊆ affected test cases
SRT: How it works

Mapping Test case <-> Code
- Create trace during test execution
- Granularity varies

Determine Modifications
- W.r.t. last tested version
- Source / binary

Identify affected test cases
- Based on traces and diffs
- Different algorithms applied

Assumption underlying safe SRT:
Test execution is unchanged except for the modifications

Effort Reduction Potential

**RQs:** How large are the savings achievable through SRT?

**Study Design:**

- Test case affected if it traverses a modified method
- Diff computation on binary level; custom Java tracer
- Computed on history of 12 snapshots

**Study Object:** System at Wincor Nixdorf, Client-server, several MLoC Java
Results

System History
- Only changes to \( S_6, S_8 \) und \( S_9 \) affected covered methods
- 41/12/30 changed
- 13/4/2 covered
- 62%/62%/6% Tests affected

In General
- Random method selection
- Same 53 test traces

Average: modifying 3 covered methods affects >50% of the test cases
Applicability of safe SRT

RQ: Are test cases deterministic? (required for SRT assumption)

Design:
• Traced $T_1$, $T_2$, $T_3$ each 6x
• Each consistent to test spec
• How many methods are covered always / only sometimes?

Result:
• In worst case: only 1/3 of methods always covered

Safe SRT techniques not applicable
How can SRT help in practice?

Semi-automated approach
- SRT proposes initial test set
- Tester manually reduces set
  ⇒ SRT reveals dependencies

Test-Gap Analysis
- SRT tooling computes changes and test coverage
  ⇒ Identify untested changes to spot missing test cases
Conclusion

Selective Regression Testing
• Proposes test cases based on changes to a system under test

Our study showed that, for manual system tests
• Reductions we can expect are small (<50% for ≥3 modified methods)
• Manual tests are not deterministic. Safe SRT approaches are thus not applicable, since test control assumption is violated

But…
• SRT can serve as helpful input for manual test selection
• SRT Tooling can help improve test control by revealing untested changes

Future Work
Employ semi-automated approach and test-gap analysis at Wincor Nixdorf