Test Case Prioritization: A Family of Empirical Studies

S. Elbaum, A. G. Malishevsky, G. Rothermel

Presented By:
Harsh Nanda
26th November, 2002

Test Case Prioritization Problem

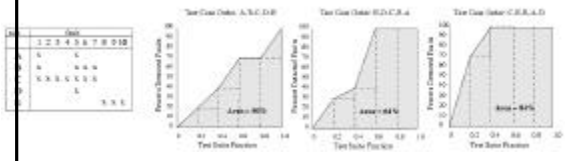
- Given:
  - T : Test suite
  - PT : Set of permutations of T
  - f : Function from PT to real numbers
- Problem:
  - Find T' in PT, such that

$$(?\,T'')(T''?\ PT)(T''?\,T')[\,f(T')\,?\ f(T'')]$$

Goal of Prioritization

- *Rate of fault detection*
- Rate of code coverage
- Rate of increase of confidence in reliability
- Rate of fault detection in specific code changes

Average of the Percentage of Faults Detected (APFD)



Test Case Prioritization Techniques

- Comparator (2)
  - Experimental controls
- Statement Level (4)
  - Fine granularity techniques
- Function Level (12)
  - Coarse granularity techniques

Comparator Techniques

- Random ordering
  - Lower bound on performance
- Optimal Ordering
  - Upper bound on performance

Statement & Function Level Techniques

3	st-total	prioritize on coverage of statements
4	st-addtl	prioritize on coverage of statements not yet covered
5	st-fep-total	prioritize on probability of exposing faults
6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
7	fn-total	prioritize on coverage of functions
8	fn-addtl	prioritize on coverage of functions not yet covered
9	fn-fep-total	prioritize on probability of exposing faults
10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
11	fn-fi-total	prioritize on probability of fault existence
12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous te
13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o
15	fn-diff-total	prioritize on probability of fault existence
16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous te
17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o

Keywords - Total v/s Additional

- Total
  - Ordering independent of execution
  - Ordering finalized before execution starts
- Additional
  - Ordering based on feedback
  - Ordering changed on the fly

Statement & Function Level Techniques

3	st-total	prioritize on coverage of statements
4	st-addtl	prioritize on coverage of statements not yet covered
5	st-fep-total	prioritize on probability of exposing faults
6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
7	fn-total	prioritize on coverage of functions
8	fn-addtl	prioritize on coverage of functions not yet covered
9	fn-fep-total	prioritize on probability of exposing faults
10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
11	fn-fi-total	prioritize on probability of fault existence
12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous te
13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o
15	fn-diff-total	prioritize on probability of fault existence
16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous te
17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o

Keywords - FEP

Fault Exposing Probability

- For every t (test case) in T (test Suite) and for every component c in program P
  - **Calculate ms(c,t)**: ratio of mutants exposed by t to the total mutants of component c
  - **Calculate "*award value*" of t**: sum of m(s,t) for every component c of program P

Statement & Function Level Techniques

3	st-total	prioritize on coverage of statements
4	st-addtl	prioritize on coverage of statements not yet covered
5	st-fep-total	prioritize on probability of exposing faults
6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
7	fn-total	prioritize on coverage of functions
8	fn-addtl	prioritize on coverage of functions not yet covered
9	fn-fep-total	prioritize on probability of exposing faults
10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
11	fn-fi-total	prioritize on probability of fault existence
12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous te
13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o
15	fn-diff-total	prioritize on probability of fault existence
16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous te
17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o

Keywords - FI

Fault Index

- Each function is assigned a fault index representing the fault proneness based on:
  - Function complexity
  - Complexity of changes introduced in the function
- For each test case, its "*award value*" is the sum of the FI values of the functions that the test case executes

Statement & Function Level Techniques

3	st-total	prioritize on coverage of statements
4	st-addtl	prioritize on coverage of statements not yet covered
5	st-fep-total	prioritize on probability of exposing faults
6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
7	fn-total	prioritize on coverage of functions
8	fn-addtl	prioritize on coverage of functions not yet covered
9	fn-fep-total	prioritize on probability of exposing faults
10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
11	fn-fi-total	prioritize on probability of fault existence
12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous te
13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o
15	fn-diff-total	prioritize on probability of fault existence
16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous te
17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o

Keyword - Diff

Syntactic Differences

- Simple alternative to Fault Index
- For each function in P & P' measure the degree of change by adding the number of lines inserted, deleted or changed
- For each test case, its "*award value*" is the sum of the Diff values of the functions that the test case executes

Statement & Function Level Techniques

3	st-total	prioritize on coverage of statements
4	st-addtl	prioritize on coverage of statements not yet covered
5	st-fep-total	prioritize on probability of exposing faults
6	st-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
7	fn-total	prioritize on coverage of functions
8	fn-addtl	prioritize on coverage of functions not yet covered
9	fn-fep-total	prioritize on probability of exposing faults
10	fn-fep-addtl	prioritize on probability of faults, adjusted to consider previous test cases
11	fn-fi-total	prioritize on probability of fault existence
12	fn-fi-addtl	prioritize on probability of fault existence, adjusted to consider previous te
13	fn-fi-fep-total	prioritize on combined probabilities of fault existence and fault exposure
14	fn-fi-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o
15	fn-diff-total	prioritize on probability of fault existence
16	fn-diff-addtl	prioritize on probability of fault existence, adjusted to consider previous te
17	fn-diff-fep-total	prioritize on combined probabilities of fault existence and fault exposure
18	fn-diff-fep-addtl	prioritize on combined probabilities of fault existence/exposure, adjusted o

Empirical Study: Motivation

- RQ1: Can prioritization improve the rate of fault detection?
- RQ2: Fine granularity or coarse granularity prioritization techniques?
- RQ3: Use or not, of predictors of fault proneness for prioritization?

Experiment Instrumentation

- Programs
  - 8 C programs (Siemens & Space)
- Versions
  - First order & higher-order versions
- Test Suites
  - Randomly select test cases from test pool
  - Stopping Criteria: Branch Coverage
  - 50 test suites for each program

Experiment 1: Prioritization

- Two experiments: statement level techniques (1a) and function level (1b) techniques
- APFD value calculations for eight levels (one per program), with 29 versions and 50 test suites per program and all prioritization techniques
- Statistical calculations to determine significance of difference in means

Experiment 1a: Results

Grouping	Means	Techniques
A	80.733	st-fep-addtl
B	78.867	st-fep-total
B	78.178	st-total
C	76.077	st-addtl

Experiment 1b: Results

Grouping	Means	Techniques
A	77.453	fn-fep-addtl
A	76.957	fn-fep-total
A	76.928	fn-total
B	73.465	fn-addtl

Experiment 2: Granularity Effects

- Pair-wise analysis of corresponding pairs
- Results

Grouping					Means	Techniques
A					80.733	st-fep-addtl
B					78.867	st-fep-total
B	C				78.178	st-total
	C	D			77.453	fn-fep-addtl
		D	E		76.957	fn-fep-total
		D	E		76.928	fn-total
			E		76.077	st-addtl
F					73.465	fn-addtl

Experiment 2: Granularity Effects

- Pair-wise analysis of corresponding pairs
- Results

Grouping					Means	Techniques
A					80.733	st-fep-addtl
B					78.867	st-fep-total
B	C				78.178	st-total
	C	D			77.453	fn-fep-addtl
		D	E		76.957	fn-fep-total
		D	E		76.928	fn-total
			E		76.077	st-addtl
F					73.465	fn-addtl

Experiment 2: Granularity Effects

- Pair-wise analysis of corresponding pairs
- Results

Grouping					Means	Techniques
A					80.733	st-fep-addtl
B					78.867	st-fep-total
B	C				78.178	st-total
	C	D			77.453	fn-fep-addtl
		D	E		76.957	fn-fep-total
		D	E		76.928	fn-total
			E		76.077	st-addtl
F					73.465	fn-addtl

Experiment 2: Granularity Effects

- Pair-wise analysis of corresponding pairs
- Results

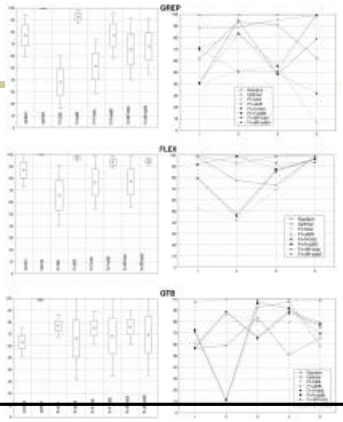
Grouping					Means	Techniques
A					80.733	st-fep-addtl
B					78.867	st-fep-total
B	C				78.178	st-total
	C	D			77.453	fn-fep-addtl
		D	E		76.957	fn-fep-total
		D	E		76.928	fn-total
			E		76.077	st-addtl
F					73.465	fn-addtl

Experiment 3: Adding Prediction of Fault Proneness

Grouping	Means	Techniques
A	79.479	fn-diff-fep-addtl
A	79.450	fn-diff-fep-total
A B	78.712	fn-fi-fep-total
B C	78.167	fn-fi-fep-addtl
C D	77.453	fn-fep-addtl
C D	77.321	fn-fi-total
C D	77.057	fn-diff-total
D	76.957	fn-fep-total
D	76.928	fn-total
E	74.596	fn-fi-addtl
E	73.465	fn-addtl
F	67.666	fn-diff-addtl

Case Study: Objects of Study

- Grep & Flex – 5 versions each
  - Publicly available v/s test suite/fault data
  - Test suite: Category Partition Method
  - Faults: Manual seeding (1/20 % criteria)
- QTB – 6 versions (300K LOC)
  - Test execution takes 27 days!
  - 139 test cases & 69% function coverage
  - Coverage information only functional



Case Study: Results - Surprises!

- QTB: Mean APFD values for feedback based techniques is lesser than the mean APFD values of techniques which do not use feedback
- Flex & Grep: Random prioritization performs better than most heuristics
- Techniques using fault proneness did not produce substantial improvements

Conclusion

- Statistically significant improvements
  - Greater variance in case studies
  - Vary across programs
- Function level techniques quite close to statement level techniques
- Statistically significant but small & inconsistent improvements using fault proneness

Real World

- Cost of prioritization v/s savings
- Saving Factor (SF)
  - Cost of executing test suite
  - Environmental factors
- C(A)-C(B) < SF*(APFD(A)-APFD(B))
  - Not taking repeated testing into account
  - Linear savings factor

I did not but you should !

- Sec 2: Test case prioritization problem
- Sec 4.2: Empirical approaches & challenges
- Sec 5.1.2: Prioritization & analysis tools
- Sec 5.3: Threats to validity
- Sec 6.2: Design