

Coverage is not strongly correlated with test suite effectiveness

Laura Inozemtseva

Reid Holmes

背景

- ソフトウェアテストは質の良いソフトウェアを提供する上で重要
 - バグを検出できるかはテストの質による
- テストについて書かれた本では、テストカバレッジ（全体のソースコードに対してテストされた割合）をテスト品質の有効なメトリクスであると書かれて場合がある

➡ テストカバレッジとテストの質に相関はあるのか？？

テストケースとテストスイート

- テストケース
 - テストの適合要件を定めたもの。一般的に実行前条件・入力値・期待結果・実行後条件の組み合わせとして定義される
 - ▶ 実際にソフトウェアのコードをテストするコード
- テストスイート
 - 目的や対象ごとに複数のテストケースをまとめたもの。自動化テストにおいては、テストの実行単位となる

先行研究

Table 1: Summary of the findings from previous studies.

Citation	Languages	Largest Program	Coverage Types	Findings
[15, 16]	Pascal	78 SLOC	All-use, decision	All-use related to effectiveness independently of size; decision is not; relationship is highly non-linear
[17]	Fortran Pascal	78 SLOC	All-use, mutation	Effectiveness improves with coverage but not until coverage reaches 80%; even then increase is small
[14]	C	5,905 SLOC	All-use, decision	Effectiveness is correlated with both all-use and decision coverage; increase is small until high levels of coverage are reached
[39]	C	<2,310 SLOC	Block	Effectiveness is more highly correlated with block coverage than with size
[24]	C	512 SLOC	All-use, decision	Effectiveness is correlated with both all-use and decision coverage; effectiveness increases more rapidly at high levels of coverage
[6]	C	4,512 SLOC	Block, c-use, decision, p-use	Effectiveness is moderately correlated with all four coverage types; magnitude of the correlation depends on the nature of the tests
[3]	C	5,000 SLOC	Block, c-use, decision, p-use	Effectiveness is correlated with all four coverage types; effectiveness rises steadily with coverage
[31]	C C++	5,680 SLOC	Block, c-use, decision, p-use	Effectiveness is correlated with all four coverage types but the correlations are not always strong
[19, 37]	C Java	72,490 SLOC	AIMP, DBB, decision, IMP, PCC, statement	Effectiveness correlated with coverage; effectiveness correlated with size for large projects
[5]	C	4,000 SLOC	Block, c-use, decision, p-use	None of the four coverage types are related to effectiveness independently of size
[20]	Java	$O(100,000)$ SLOC	Block, decision, path, statement	Effectiveness correlated with coverage across many projects; influence of project size unclear

- 先行研究では、テストスイートの質とカバレッジとの相関が示されている
- しかし、どの程度のテストケースを持っているテストスイートと相関があるかは明らかではない
 - テストスイートに含まれているテストケースの質とカバレッジとの相関は明らかになっていない

目的

- テストスイートの有効性とカバレッジの相関を確認する
 - テストスイートの中のテストケースの質とカバレッジの相関を確認する

リサーチクエスチョン

1. テストスイートの有効性は、テストスイートに含まれるテストケースの数と相関があるか？
2. テストスイートに含まれるテストスイートの数を考慮しないとき、テストスイートの有効性はカバレッジと相関があるか？
3. テストスイートに含まれるテストスイートの数を考慮したとき、テストスイートの有効性はカバレッジと相関があるか？

アプローチ

1. ツールを使いプロダクトコードに欠陥を混入させる

- PIT[35] : プロダクトコードにミュータントバグを自動生成する

2. 様々なサイズのテストスイート1000個ずつ作成する

- テストケース数 3, 10, 30, 100 ...

3. それぞれのテストスイートのカバレッジを算出する

- statement, decision, modified condition coverage

4. 平等に評価するために, テストスイートの効果を正規化して評価を行う

- カバレッジ50%と60%のテストスイートがあり, 両方とも混入されたミュータントバグの50%を検出できた場合カバレッジ50%のテストスイートの方が効果的であるため

対象プロジェクト

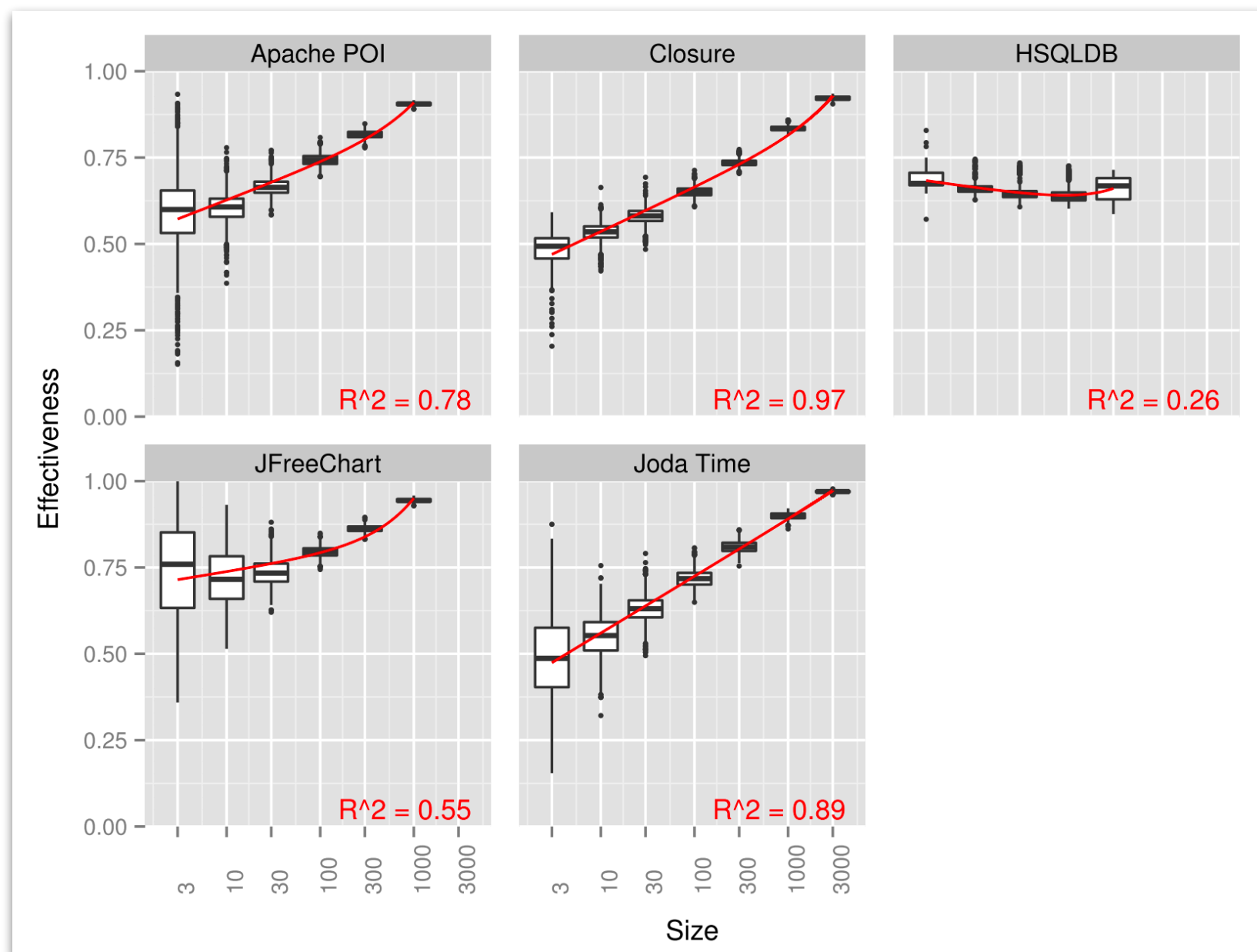
- 大規模なJavaプロジェクトを選択

Table 2: Salient characteristics of our subject programs.

Property	Apache POI	Closure	HSQldb	JFreeChart	Joda Time
Total Java SLOC	283,845	724,089	178,018	125,659	80,462
Test SLOC	68,932	93,528	18,425	44,297	51,444
Number of test methods	1,415	7,947	628	1,764	3,857
Statement coverage (%)	67	76	27	54	91
Decision coverage (%)	60	77	17	45	82
MC coverage (%)	49	67	9	27	70
Number of mutants	27,565	30,779	50,302	29,699	9,552
Number of detected mutants	17,935	27,325	50,125	23,585	8,483
Number of equivalent mutants	9,630	3,454	177	6,114	1,069
Equivalent mutants (%)	35	11	0.4	21	11

RQ1 結果

- テストスイートの有効性は、テストスイートに含まれるテストケースの数と相関があるか？



大規模なプロジェクトでは強い相関がある

RQ2結果

- テストスイートに含まれるテストスイートの数を考慮しないとき、テストスイートの有効性はカバレッジと相関があるか？

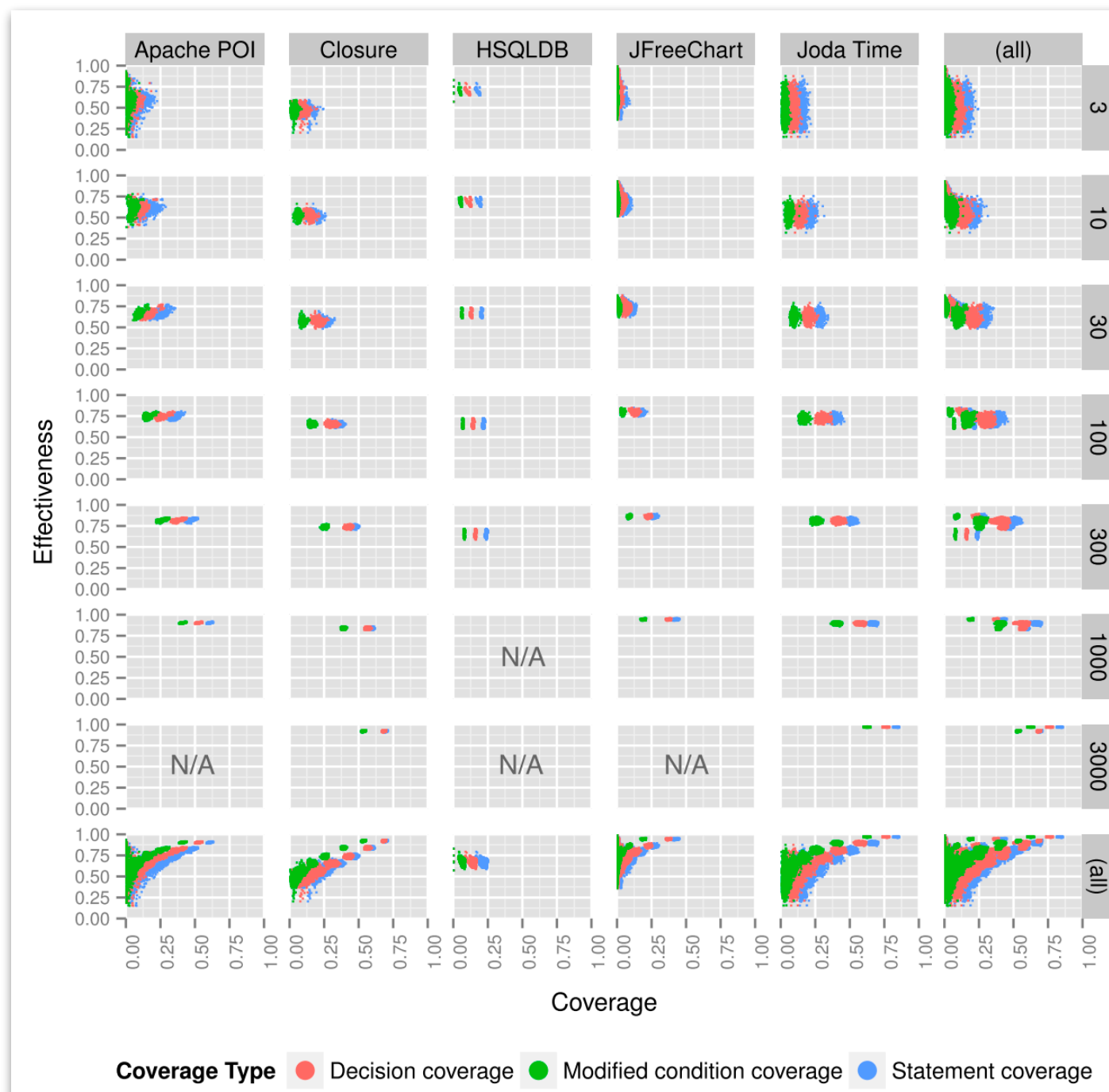
ケンドールの順位相関係数を用いて検証

Project	Statement	Decision	Mod. Cond.
<u>Apache POI</u>	0.75	0.76	0.77
<u>Closure</u>	0.83	0.83	0.84
HSQldb	-0.35	-0.35	-0.35
JFreeChart	0.50	0.53	0.53
<u>Joda Time</u>	0.80	0.80	0.80

テストスイートのサイズを考慮しない場合は、多くの大きいプロジェクトでは強い相関がある

RQ3結果

- テストスイートに含まれるテストスイートの数を考慮したとき、テストスイートの有効性はカバレッジと相関があるか？



RQ3結果

- ケンドールの順位相関係数を用いて検証すると
 - 相関の代償はプロジェクトによって様々
 - Apache POIはテストスイートのサイズを考慮していなければ、相関は0.94 考慮すると0.46~0.85
 - Joda Timeはテストスイートのサイズを考慮していなければ、相関は0.80~0.85 考慮するとほぼゼロ
- 結果がプロジェクトによって異なるため、相関があるとは一概に言えない

考察

- テストスイートのサイズを考慮しなければ、相関があるがサイズを考慮すると相関は小～中程度となるためカバレッジをテストスイートの質の指標とするのは現実できでない
- カバレッジが低いことはテストが不十分であることは示せるが、質が良いことは示せない

まとめ

- テストスイートのサイズを調整して、カバレッジとの相関を確認した
 - 一般的には、テストスイートのサイズを調整した場合は、カバレッジと弱い相関がある
 - しかし、カバレッジとの相関の強さはプロジェクトによって様々であった
 - しかし、特定のタイプのカバレッジでは強い相関を示すものもある
- 結果的には、テストスイートの効果とカバレッジの関係を強く肯定するほどの相関は得られなかった

今後の発展

- 今回はミュータントバグを使用したか、実際の不具合を利用して効果を測定する
- カバレッジとバグレポートを参照して、テストスイートでカバーできているにも関わらず不具合が出た箇所を特定し、どのような不具合が出たかを調べる
 - どのような不具合がテストで特定しにくいかが分かるかも