PNSQC

OCTOBER 10-12 2022

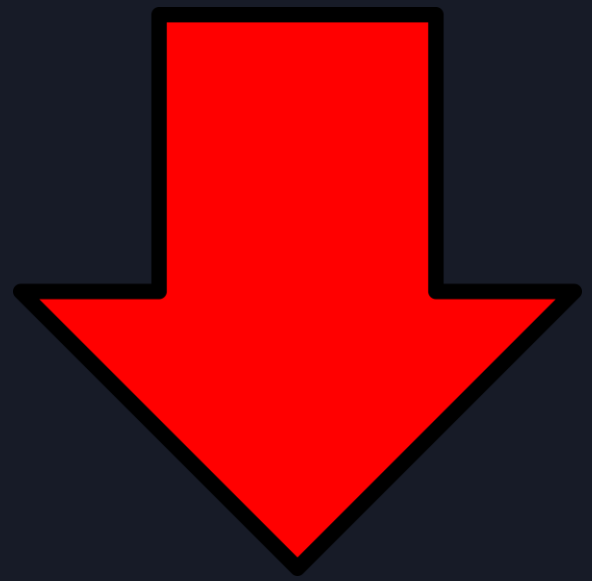
Jack Marvin & Trevor Hammock

A Method to Select Tests Based on Code Coverage

Some Questions

- What are the advantages of selecting tests based on code changes?
- What are the risks associated with code-based testing?
- Who can benefit?

Advantages of selecting tests based on code changes



- Run less tests. Average 50-80%.

- Depends on risk tolerance and confidence
- Code change frequency
- Developer experience level

Advantages of selecting tests based on code changes

- Developers get “Is my change good?” sooner
- Rate of code change limiting factor, not testing

Advantages of selecting tests based on code changes

- Focused on changes
- Earlier confidence in change
- If it did not change then low risk of fail
- Only tests related to change run

Advantages of selecting tests based on code changes

- Frees up hardware - exploratory

- QA gets the answer to “Is this build deliverable ” much sooner

- Reduce false fails

- Can be large benefit if environment not reliable

Advantages of selecting tests based on code changes

- Reduced redundant fails from the same defect

- Identify tests that haven't failed over time
  - Great for pruning regression suites

- Detect 'valuable tests', those with high coverage
  - Can prioritize high coverage tests

Advantages of selecting tests based on code changes

- Help identify which functions are most common across the tests
- An automated method to map features to tests
- Determine code churn and indicate stability

Advantages of selecting tests based on code changes

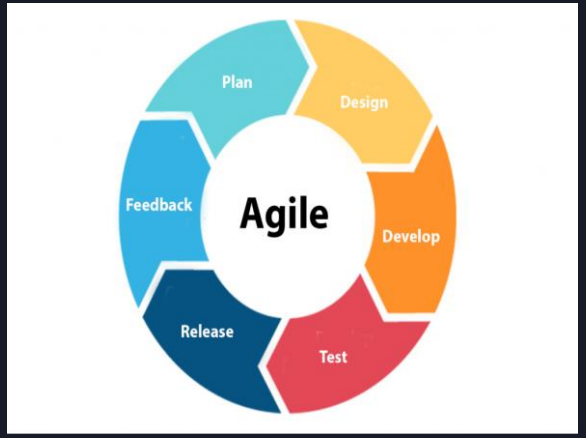
- Can quantify a changes impact
- Not supposed to have effect
- Aid other teams in flow

Risks Associated with Code-based Test Selection

- Educated guesswork for “best” tests
  - Highest coverage with fastest turnaround time
- Not running all tests may miss a defect

Risks Associated with Code-based Test Selection

- Release cycle, iterations and stability
- Number of tests varies with Milestones



- Cross-team Collaboration – Anti-Silo
- Focusing on code vs. upstream or downstream

Risks Associated with Code-based Test Selection

Ways to limit the risk

- Start slow, increasing use of reduced tests as confidence increases.
- Run full list and compare fails to reduced test list
- Delay use for key releases

Who can Benefit

Anyone can use it

- Developers
- QA
- Marketing

How Does it Work?

- Regression suite run with a code coverage tool
- Database stores mapping of code to test
- Analyze build changes
- Map changes to database gives tests

main.cpp.gcov

```
1        -:    0:Source:main.cpp
2        -:    0:Programs:3
3        -:    1:#include "mainwindow.h"
4        -:    2:#include <QApplication>
5        -:    3:
6        1:    4:int main(int argc, char *argv[])
7        -:    5:{
8        1:    6:    QApplication a(argc, argv);
9        2:    7:    MainWindow w;
10       1:    8:    w.show();
11       -:    9:
12       2:   10:    return a.exec();
13       -:   11:}
```

How Does it Work?

How much time do these steps take?

- Code coverage regression run > regular

- Database creation
- 0.5 to 2 hours, code and test size

- Test list creation 10-60 seconds

Other Limitations and Risks

- New code changes unknown to database
- Regular code coverage runs
- Need to add new tests
- Still proves no regressions

Other Limitations and Risks

What is the optimal set of tests?

- Fastest
- Least number with combined coverage
- Less tests != less runtime

Other Limitations and Risks

Common code changes gives most tests?

- Think main.C
- Add filtering with increased risk of missing a defect
- Can filter for the optimal tests

Other Limitations and Risks

Rare-occurring defects
not detected as early
- Intermittent seg faults
- Rare race condition fails

Other Limitations and Risks

- Code coverage tool targets specific language

- Functional or line coverage
- Functional less accurate
- Line changes often
- Line is difficult to uniquely identify

Conclusion

- Although not using this tool for all builds by all teams until confidence allows, no defects have been missed and we have greatly lowered the test burden.
- We hope to increase usage as benefits are shown
- Allows for new forms of testing which we are currently exploring.

PNSQC

OCTOBER 10-12 2022

A Method to Select Tests Based on Code Coverage

THANK YOU

SPEAKER NAME

TOPIC

40TH ANNUAL
PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022