

Software Testing without Test Design

**Stuart Reid
STA Consulting Inc.**

www.stureid.info / stureid.test@gmail.com

©Stuart Reid, 2017

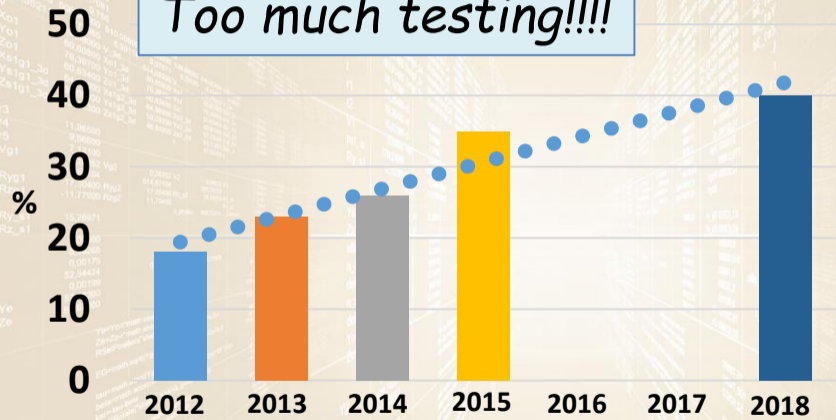


Supported by:



Testing as a Percentage of IT Budget

Too much testing!!!!



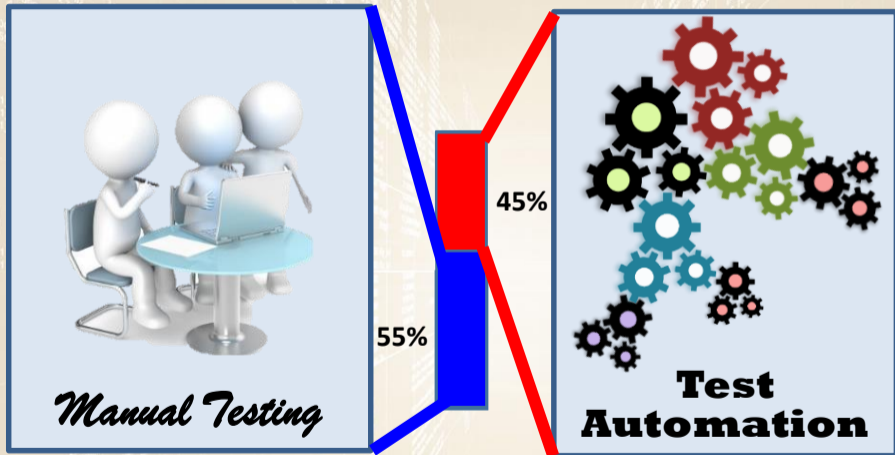
Is it good enough testing?

World Quality Report 2015-16

Automation is only a partial answer

JTC1/SC7

Interim Meeting - Mexico 2017



World Quality Report 2015-16



Supported by:



Testing WITHOUT Test Design

- **Tests generated by the end users**
 - crowd testing and A/B testing
- **Tests generated using random test generation**
 - pure random and fuzz testing
- **Tests generated by artificial intelligence**
 - regression tests and stress tests



Users as Testers - Crowd Testing & A/B Testing

What about the users as testers?

- **Their personas are perfect 😊**
 - we don't have to guess their habits or preferences
- **Their test environments are truly representative 😊**
- **Can we trust them? 😞**
 - use Non-Disclosure Agreements (NDAs)
 - don't tell them!



Mobile

JTC1/SC7

Interim Meeting - Mexico 2017



Supported by:

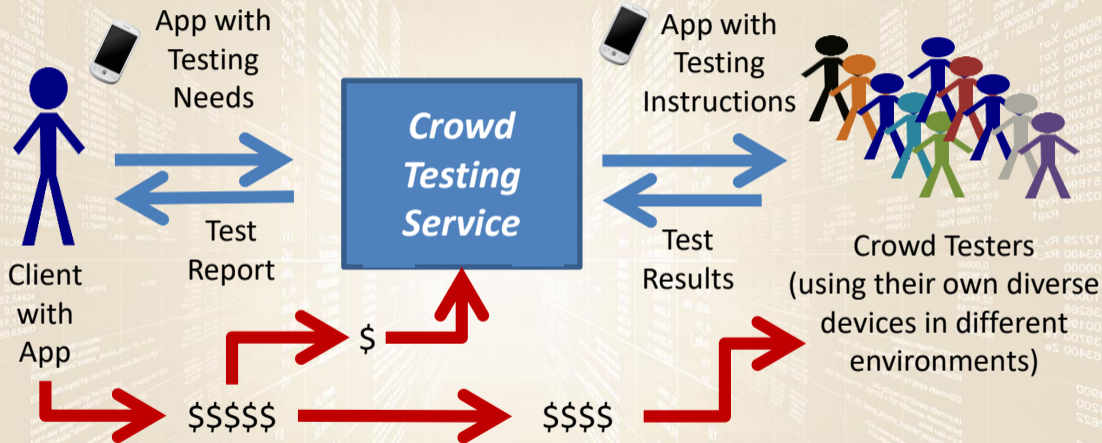


Android Handset Fragmentation

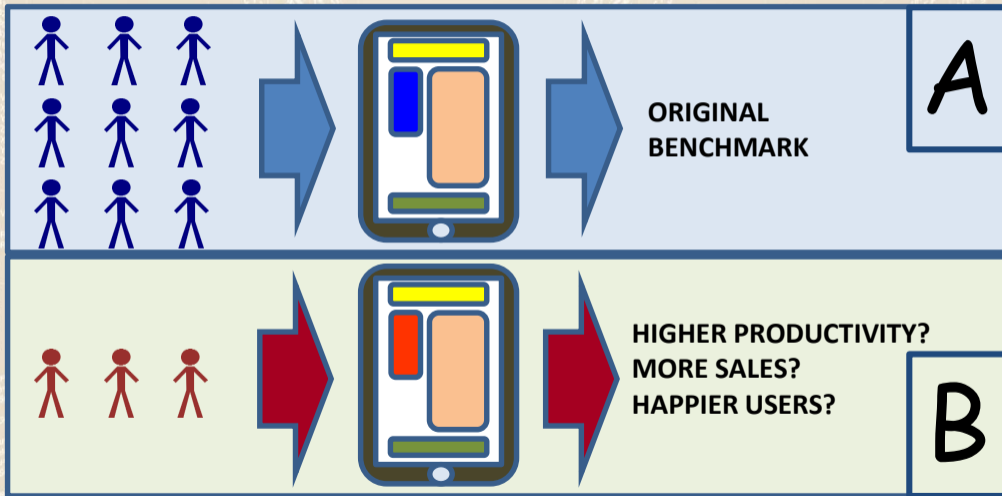
24,000 distinct
devices
(Aug 2015)



Crowd Testing for Multiple Devices/Env'ts

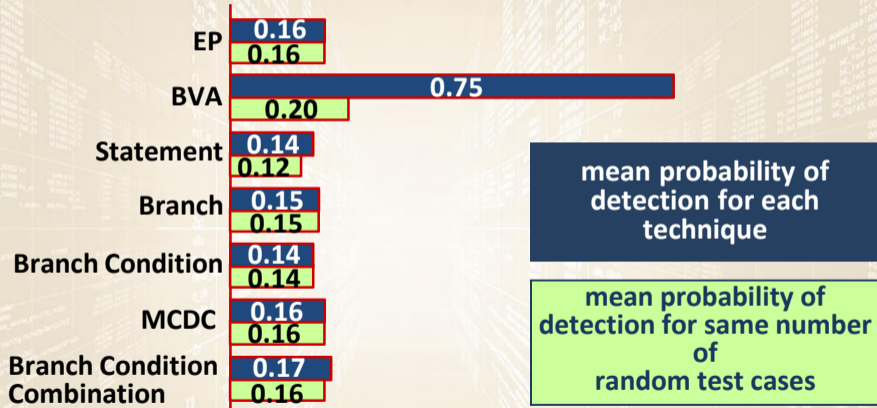


A/B Testing



Random Testing & Fuzz Testing

Experimental Evidence— Random vs Systematic Test Design



Big Data - Example Data Volumes

JTC1/SC7

Interim Meeting - Mexico 2017

- **Airliner**
 - 3 GB/hour
- **CCTV**
 - 1 TB/month
- **Mobile Phone Operator**
 - 3 TB/day
- **Globally**
 - 2.5 Million TB/day



Supported by:



Data Validity from Internet of Things

We're drowning!!
There's too much data

Interface
Standard

Keep me safe!!
Stop insecure data

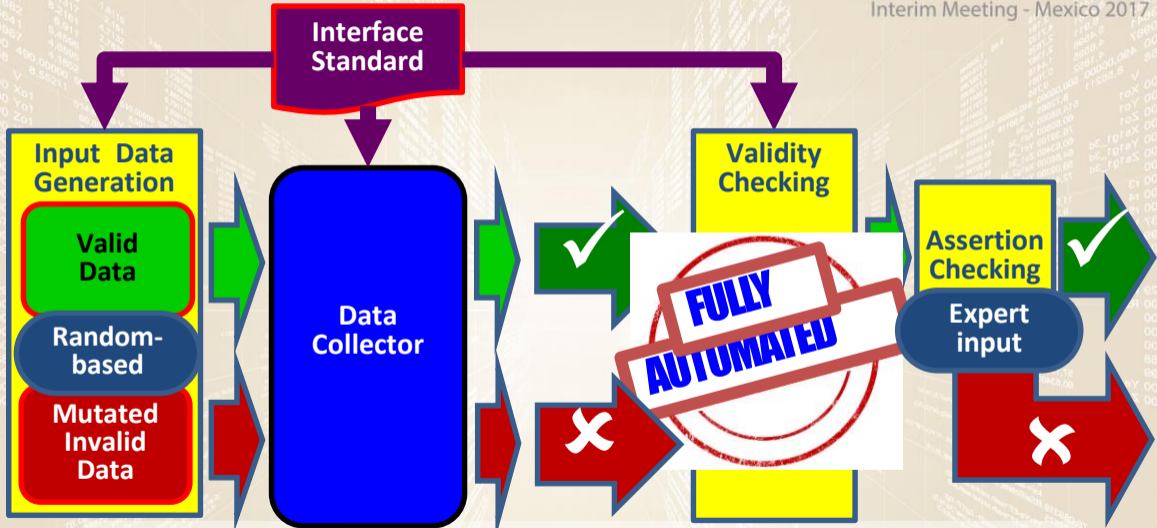
Raw
Sensor
Data

Data
Collector

Big Data

Analytics

Fuzz Testing

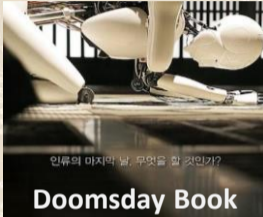


Testing with Artificial Intelligence

Artificial Intelligence in the Cinema

JTC1/SC7

Interim Meeting - Mexico 2017



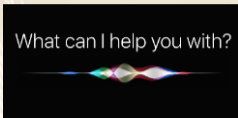
Supported by:



Artificial Intelligence (AI) Works!

JTC1/SC7

Interim Meeting - Mexico 2017



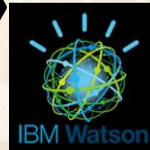
Supported by:



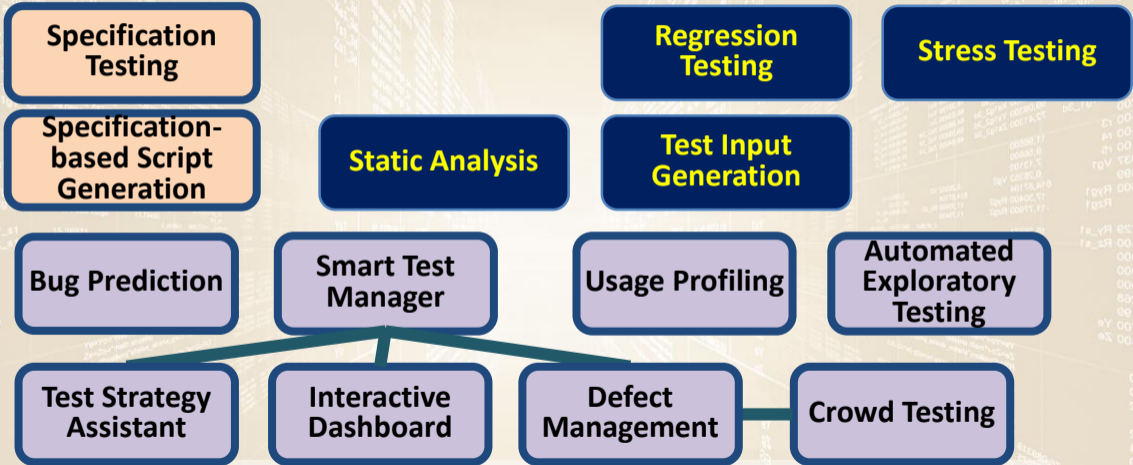
Artificial Intelligence Techniques

- **Neural Networks**
- **Expert/Knowledge-based Systems**
- **Machine Learning**
- **Fuzzy and Probabilistic Logic**
- **Classification**
- **Search and Optimization**
- **Much of today's effective AI uses a variety of overlapping techniques**
 - and exploits the availability of processing power & storage

AI Toolkits



AI - Smart Testing Opportunities



Static Analysis

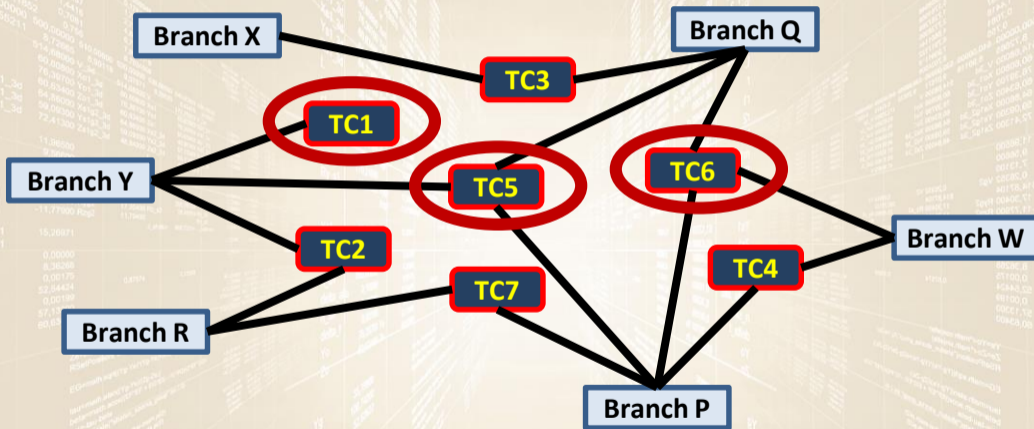
Static Analysis - Facebook Infer

- **Open source**
- **Analyses C, Objective-C and Java**
 - on Android and iOS
- **Fast – can do millions of LOC in a few minutes**
 - ideal for continuous integration
- **Facebook claims that approximately 80% of raised issues are fixed (so are true faults)**
- **Also used by Instagram, Uber, Spotify, etc.**

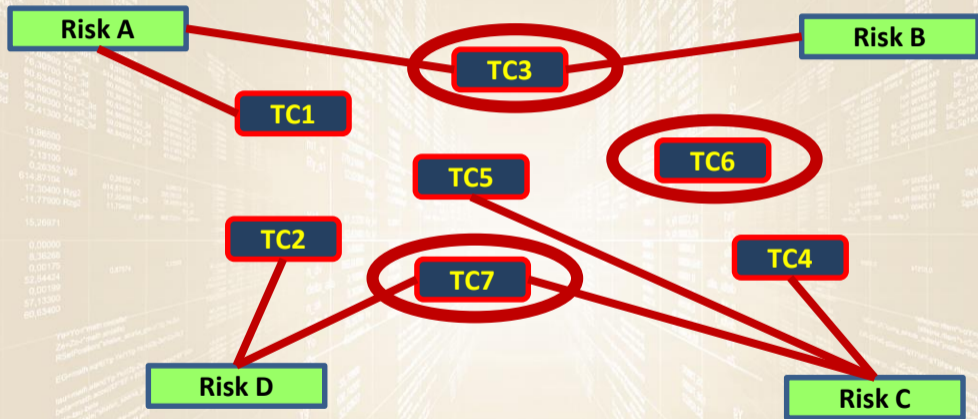


Regression Testing

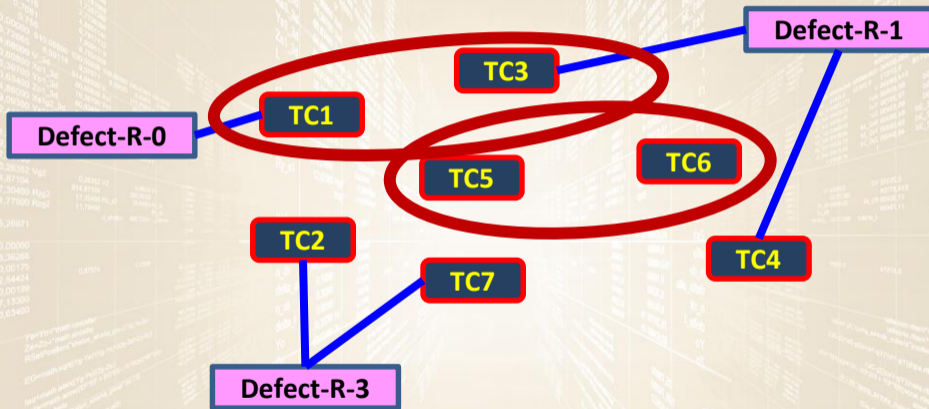
Regression Test Optimization



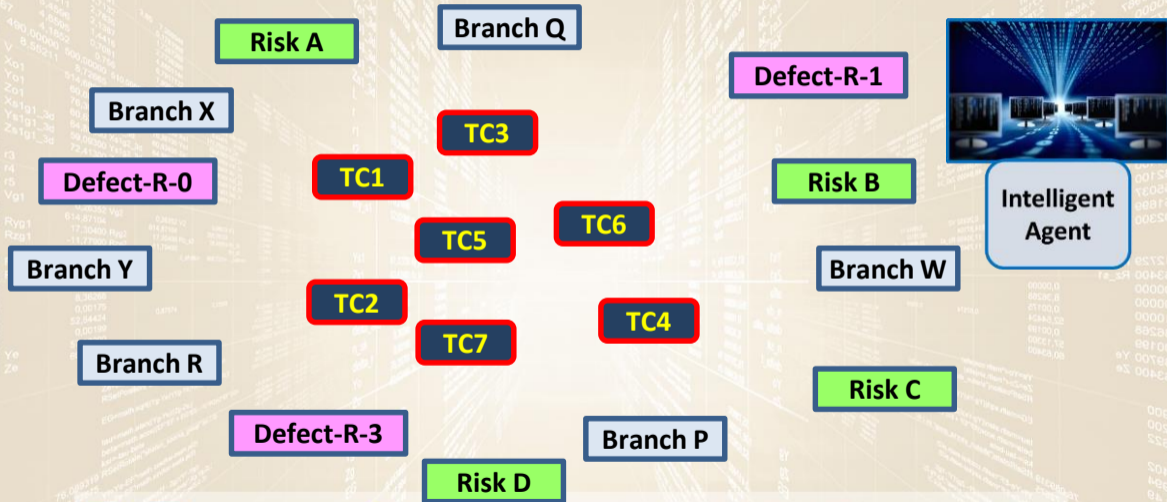
Regression Test Optimization



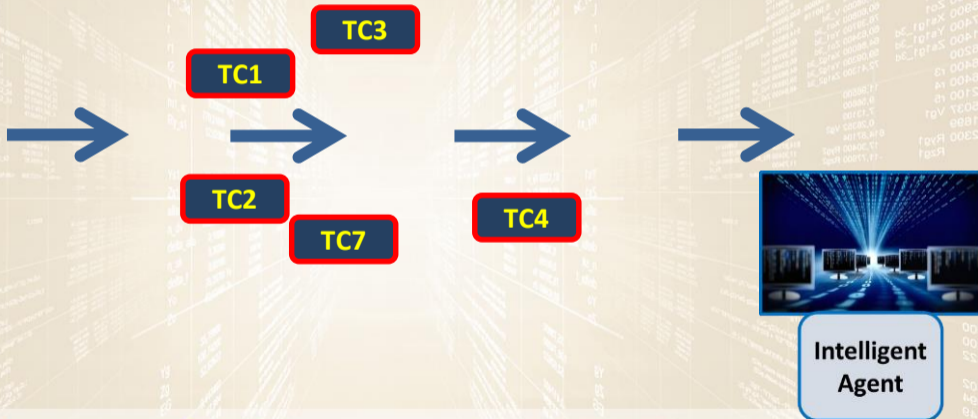
Regression Test Optimization



Regression Test Selection



Regression Test Prioritization



Regression Test Optimization Criteria



Interim Meeting - Mexico 2017

- Tests that found defects previously
- Tests that reduce execution time
- Reduce the number of tests needed
- Tests that achieve full coverage
- Test that exercise recently changed code
- Tests that address high risks
- etc.



Supported by:

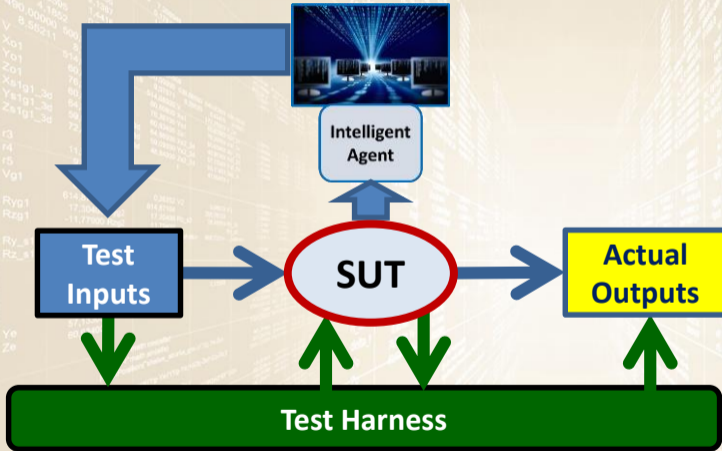


Regression Test Optimization Results

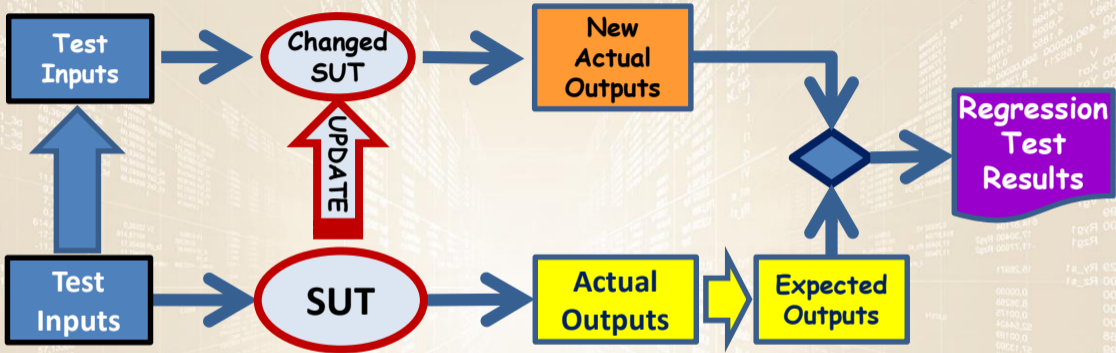
- **The algorithm reduces the test suite data by approximately 50%**
 - [Rai, 2014]
- **The techniques are 40-50% more effective in uncovering the first failure of the changed program**
 - [Jiang, 2009]
- **Average reduction in test suite size of 94% while achieving requirements-based coverage**
 - implemented in:
 - a continuous integration env't with 30 seconds run time
 - implemented at Cisco, Norway
 - [Gotlieb, 2016]

Automated Test Input Generation

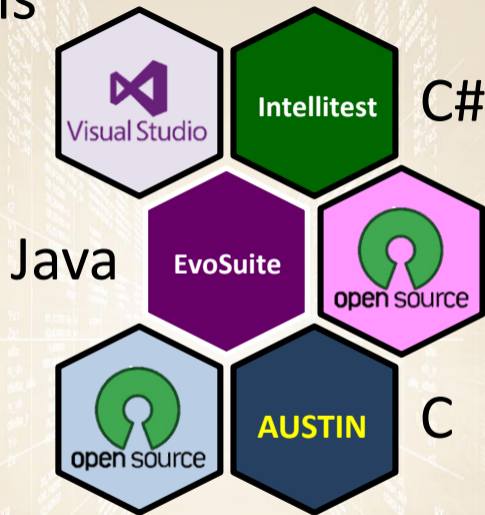
Automated Test Input Generation



Automated Test Input Generation



Example Tools



Automated Test Input Generation - Summary

- **Empirical studies have shown:**
 - tool support can lead to improvements in code coverage of up to 300%
 - that there is no measurable improvement in the number of bugs actually found by developer/testers – even though more branches are covered
- **A set of automatically-generated regression tests providing full coverage is an excellent starting point when you change or refactor the code**
- **Danger!!!**
 - testers rely on the tool → little or no black box testing
 - testers use the tool to meet safety-related test standards

Automated Stress Testing

Automated Stress Testing Tools

- **Generate pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events**
 - they pretend they are a ‘stupid’ tester
- **Aim to cause an ANR (‘Application Not Responding’) or for the app to simply crash**
 - so test result is easy to observe
- **Require little tester input**
 - except to check-out the reported failures

Example - Android Stress Testing Tools

- **Google Monkey**

- built into the Android development platform - free
- fuzz testing tool – random inputs

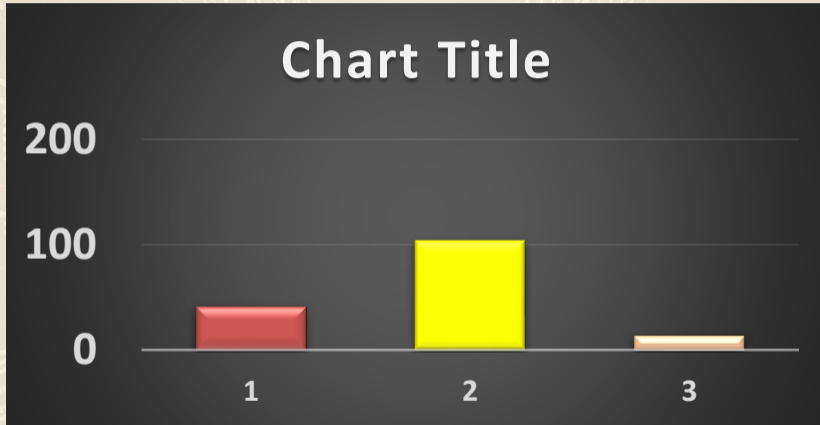
- **Sapienz**

- open source
- search-based testing tool
- when applied to the top 1,000 Google Play apps, Sapienz found 558 unique, previously-unknown faults

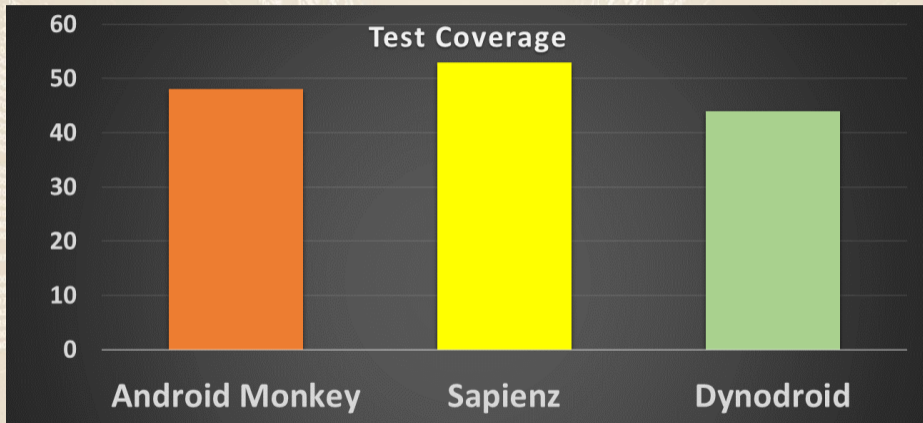
- **Dynodroid**

- open source
- allows interleaving of human and tool
- when applied to the top 1,000 Google Play apps, Dynodroid found 6 unique, previously unknown faults

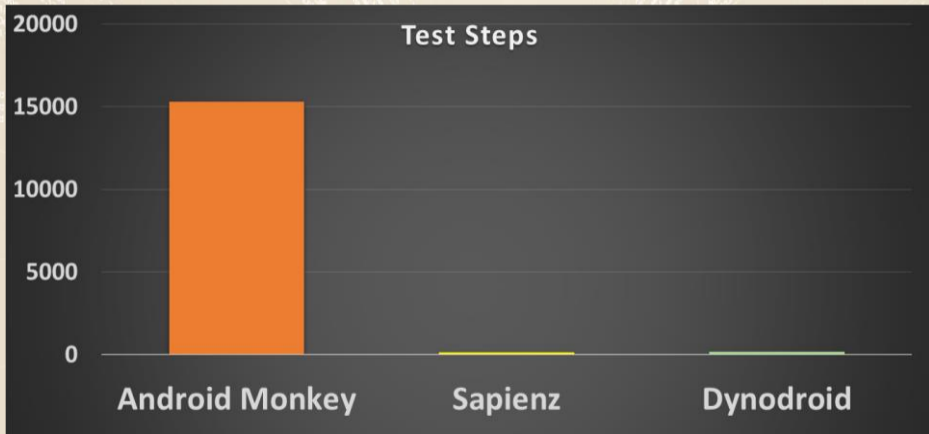
Defect Detection Effectiveness



Test Coverage



Fault Revealing Steps



Conclusions

- **Testing must get more efficient**
 - we can't justify spending 40% of the IT budget
 - there aren't enough professional testers to spend this on
- **We have to think 'outside of the (black and white) box'**
- **We need to supplement traditional approaches...**

Testing WITHOUT Test Desi

Thank you for listening 😊

Any Questions?