

THE PATH TO

CONTINUOUS TESTING

Brian Kitchener

1



INERTIAL
CONSULTING
SOFTWARE TESTING SOLUTIONS

Brian Kitchener - Founder and Consultant
brian@inertialconsulting.com
<http://www.inertialconsulting.com>
<http://mrselenium.blogspot.com>

2

WHAT IS CONTINUOUS TESTING?

WHAT IS CONTINUOUS TESTING?

3

WHAT IS CONTINUOUS TESTING?

Automated Tests Executed Continuously

WHAT IS CONTINUOUS TESTING?

4

THAT DOESN'T SOUND QUITE RIGHT???

SOME QUESTIONS

- ▶ What is the difference between continuous integration and continuous testing?
- ▶ What is the difference between automated testing and continuous testing?
- ▶ What's the difference between continuous deployment and continuous testing?
- ▶ What value does continuous testing provide?

5

WHAT IS

CONTINUOUS TESTING?

6

WHAT IS CONTINUOUS TESTING?

- ▶ Continuous testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate.
—wikipedia

7

WHAT IS CONTINUOUS TESTING?

- ▶ Continuous testing is the process of executing **automated tests** as part of the **software delivery pipeline** to obtain immediate **feedback** on the **business risks** associated with a software release candidate.

8

WHAT IS CONTINUOUS TESTING?

- ▶ automated tests
software delivery pipeline
feedback business risks

9

CONTINUOUS TESTING IS ABOUT UNDERSTANDING

RISK

10



THE STORY OF

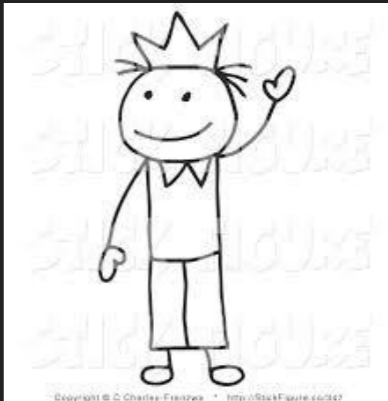
TODD

a story that definitely didn't happen

11

THIS 'TODD'

- ▶ Todd is a great cook
- ▶ Todd loves animals
- ▶ Todd enjoys LARPing on the weekend
- ▶ Todd thinks he's brilliant
- ▶ Todd is a terrible developer



12

OK MAYBE I AM A LITTLE TODD

TODD IS MODIFYING A SERVICE

- ▶ It's Friday afternoon.
- ▶ Todd is fixing a defect
- ▶ While refactoring he discovers some "bad" code and duplicate calls
- ▶ Being "brilliant" Todd cleans up the service and removes the sloppy code
- ▶ Todd doesn't know the "sloppy" code was implemented as part of a fix for an edge-case defect within the caching layer
- ▶ Todd's code gets deployed to the test environment

13

TODD'S FIX IS DEPLOYED

- ▶ QA (being diligent, brilliant, and extremely handsome) runs their automated regression and catches the issue
- ▶ "Regression is failing, defect logged"
- ▶ After working all weekend, the issue is identified, fixed, and the release passes regression

14

ASADENDING

- ▶ The release ended up being held up for a low priority defect that only occurs occasionally
- ▶ The business wasn't able to make a judgement call because all they knew is 'the regression is failing'
- ▶ Everyone had to work over the weekend to meet a key marketing deadline
- ▶ A bunch of people quit because "this stuff happens all the time"

15



**SO WHAT ABOUT
TODD?**

**Exhausted from working late, he was
tragically killed in a LARPing accident.**

16

ALTERNATELY....

NABETTERMODEL

- ▶ Developers know immediately when they break something
- ▶ All the automated tests map to business requirements
- ▶ At any point in time the business knows the status of the release.
- ▶ No longer does QA have to “sign-off” on a release, as the quality can be quantified
- ▶ QA doesn’t have to manually shepherd a release through each environment
- ▶ Todd lives!

17

SO WHAT IS IT???

CONTINUUSTESTINGS NOT

- ▶ Kicking off suites of tests manually
- ▶ Automating features after they are released
- ▶ Additional manual regression testing
- ▶ Manual report analysis
- ▶ Fragile tests or inconsistent results
- ▶ Asking QA if the release is ready

CONTINUUSTESTINGS


- ▶ Automated validation of business requirements
- ▶ Easily understandable status reports
- ▶ Targeted notifications on failures
- ▶ Small, independent automated tests
- ▶ Stable, trustable automated tests
- ▶ Automatic test execution through CI
- ▶ Always knowing which features are working

18

**THAT DOESN'T SEEM TOO
HARD LET'S TRY IT**

FamousLastWords

19



THE FIRST BARRIER

GETTING EVERYONE LINED UP

20

WAT...WEREDONGWHAT?
WHENDDWEDECODETHAT?

Literally Everyone

21

ORGANIZATIONAL ALIGNMENT

- ▶ Meet with key stakeholders across the organization:
 - ▶ Architecture
 - ▶ Development
 - ▶ Quality Assurance
 - ▶ Infrastructure (Ops)
 - ▶ Product & Business Org
 - ▶ Executives

22

THE VALUE

EXPLAIN COST AND BENEFITS

- ▶ Costs
 - ▶ Process Changes
 - ▶ Development costs
 - ▶ Resource Training
 - ▶ Documentation / Meetings
 - ▶ Have to keep tests up to date
- ▶ Value:
 - ▶ Improved environment stability
 - ▶ Reduced development time
 - ▶ Reduced application release time / cost
 - ▶ Lower release risk
 - ▶ Improved agility

If all else fails just tell them it's what Apple/Google/Amazon does

23



Requirements

THE SECOND BARRIER

DEB NEW WHAT TO TEST

24

IF EVERYTHING IS A PRIORITY,
NOTHING IS

Somebodyfamous

25

BUSINESS REQUIREMENTS

- ▶ Define and **prioritize** which business requirements to automate
 - ▶ Mandatory, high, low, none
- ▶ Include automated tests as part of the 'definition of done'
 - ▶ What does this mean if they aren't done or are failing?
- ▶ Decide as a team what happens when automated tests are incomplete or failing
- ▶ Include automated testing as part of the sprint planning and estimation process

26

WHAT IS BETTER?

1 TEST THAT COVERS
1000 REQUIREMENTS

1000 TESTS THAT EACH
COVER 1 REQUIREMENT

27

SMALL TESTS > BIG TESTS

- ▶ Ideally each automated test should map to a single requirement
 - ▶ Failing tests don't block each other
 - ▶ Very clearly shows what is broken, and what still works
- ▶ Provides a mechanism to scale (multi-threading)

28

CONSISTENCY
IS 

THE THIRD BARRIER

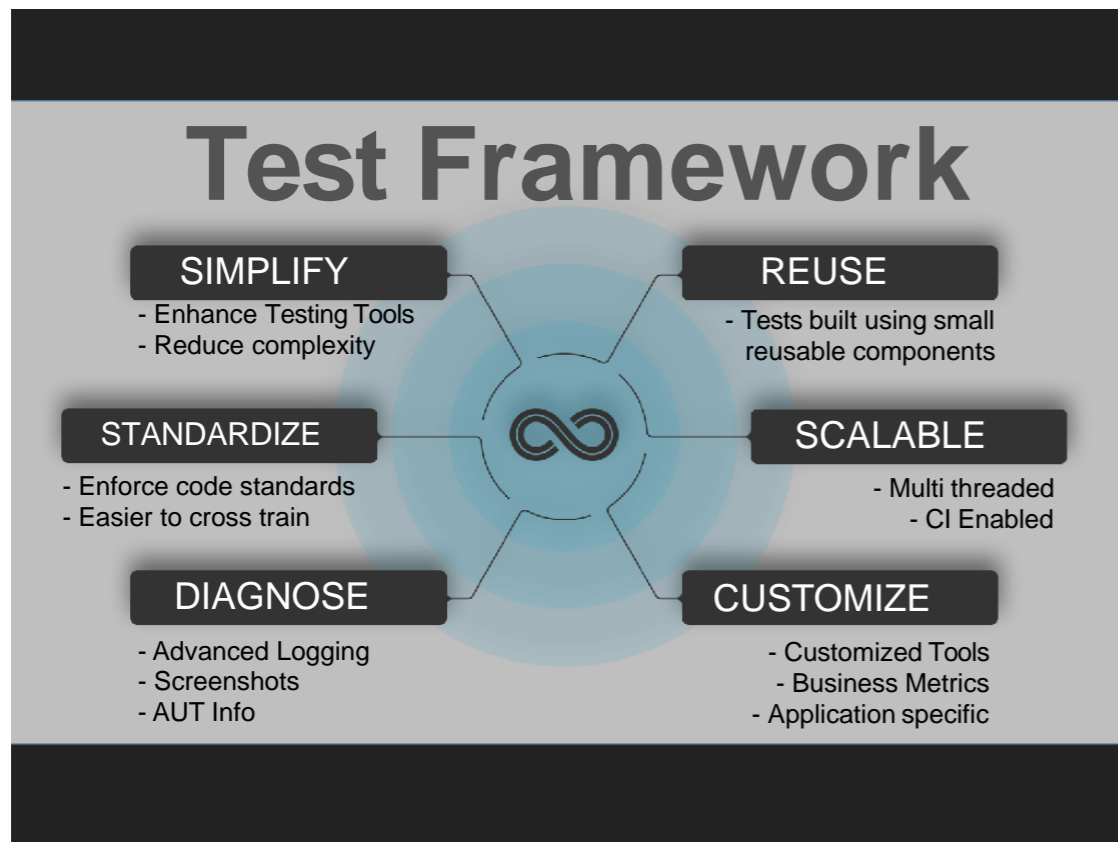
RELIABLE TESTS

29

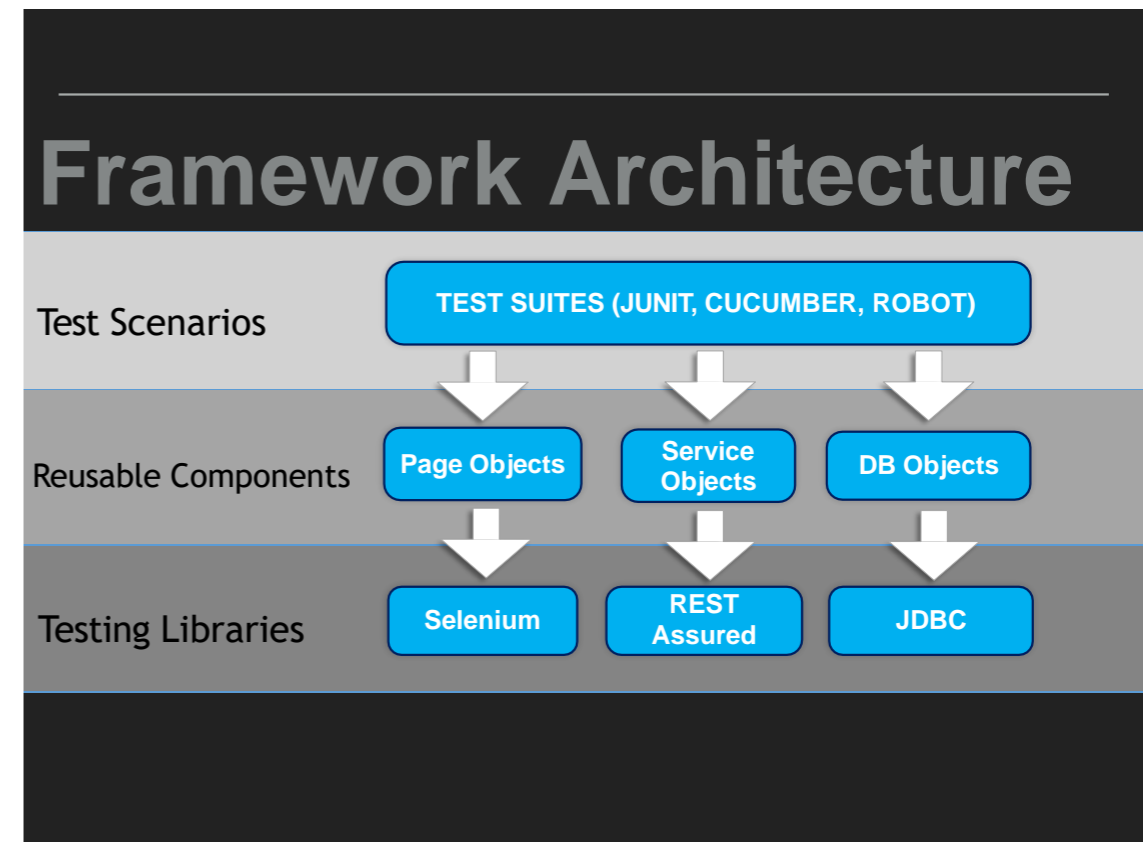
AUTOMATED TESTING SUCKS

- ▶ Testing tools are inconsistent, hard to use, and specialized
- ▶ Invest in a framework:
 - ▶ An internally developed library that can be shared across the organization
 - ▶ Any tool, any language, any technology
 - ▶ UFT, Selenium, TestComplete, JMeter
 - ▶ UI, Web, API, DB, etc
 - ▶ Java, C#, Python, Ruby, JavaScript, Go

30



31



32



THE FOURTH BARRIER

DATA STAGING

33

CONSISTENT AND REPEATABLE DATA IS KEY

- ▶ Tests should stage their own test data
 - ▶ DB sync's can change over time
 - ▶ Test data needs may change sprint to sprint
 - ▶ Continuous testing will consume a LOT of data
 - ▶ As test suites grow cross-test data issues will appear
- ▶ Alternate strategies may be necessary
 - ▶ Transactional data can't always be generated on demand
 - ▶ May not have access

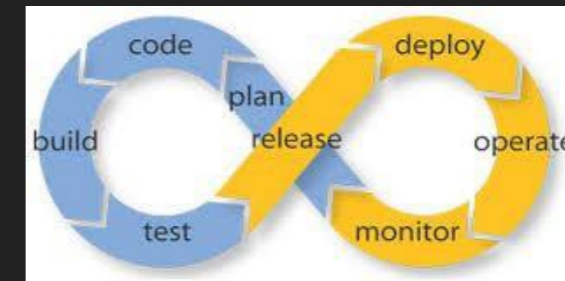
34

API IS OPTION #1

DATA STAGING

- ▶ Api's can provide easy ways to stage data
- ▶ UI data staging is slow and time consuming
- ▶ DB data staging can bypass important business logic

35



THE FIFTH BARRIER

CONTINUOUS INTEGRATION

36

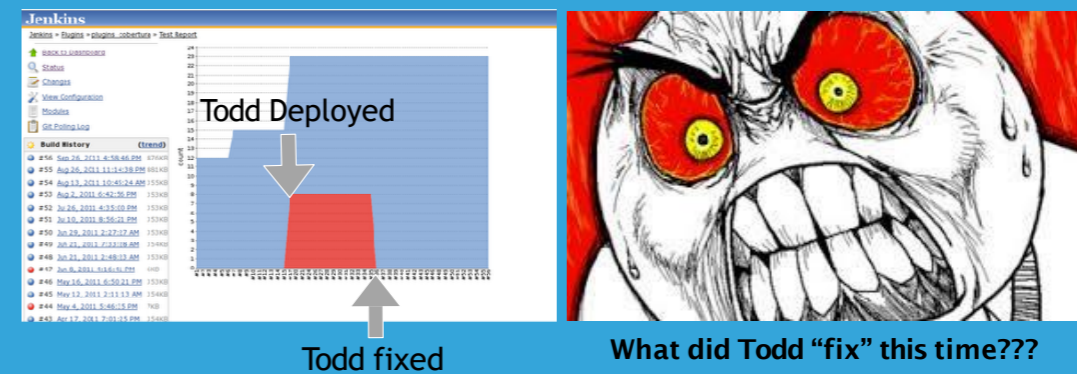
CI IS THE HEART

CONTINUOUS INTEGRATION

- ▶ Any CI server will do
 - ▶ Bamboo, Jenkins, CircleCI, TeamCity, etc
- ▶ Provides a place to execute tests, store the test results, and notify of failures.
- ▶ Start with a manual job, and add triggers when ready!
- ▶ Allows anyone or anything ability to execute the tests
- ▶ Standard XML report structure
- ▶ Get Todd a hate of shame

37

CI INTEGRATION



38



THE SIXTH BARRIER

FEEDBACK CYCLES

39

BEWARE OF SPAM

CISERVERENABLESNOTIFICATIONS

- ▶ Ci servers come with built in notifications
 - ▶ Email, IM, Text Message
- ▶ Need to have clearly defined targets
 - ▶ Notify the developers, not the automation engineers

40

PEOPLE NEED TO CARE

- ▶ “The tests are failing”
- ▶ “Yeah they have been”
- ▶ “For how long?”
- ▶ “The last few months, we’ll fix them eventually”

41

WHEN TESTS FAIL

- ▶ Failing tests need to be fixed or removed
- ▶ Impacted teams need to be notified and prioritize fixing issues
- ▶ Developers need to be able to quickly understand test failures
- ▶ Don’t just comment out the failing code!!!!

42



THE SEVENTH BARRIER

RISK IDENTIFYING TESTS

43

REQUIREMENT MAPPING

- ▶ No one cares about broken tests, they care about broken business requirements
- ▶ A requirements traceability matrix can help, but isn't as clear
- ▶ Mark or comment automated tests with business requirements
 - ▶ Cucumbertags
 - ▶ JavaAttributes
 - ▶ Comments
- ▶ Ideally each test maps to a single requirement

44

NONFUNCTIONAL REQUIREMENTS

- ▶ Security scanning
- ▶ Code Quality
- ▶ Performance Testing

45



THE FINISH LINE

DEVOPS UNLOCKED

46

LASTLY

DEVOPSUNLOCKED

- ▶ Continuous Testing enables Continuous Delivery and Deployment.
- ▶ Start small and iterate
- ▶ Pilot teams can provide proof of concept, and help onboard more teams onto the model

47

The logo for Inertial Consulting features a stylized graphic of a curved line of dots above the word "INERTIAL" in a bold, sans-serif font. Below "INERTIAL" are the words "CONSULTING" and "SOFTWARE TESTING SOLUTIONS" in a smaller, spaced-out sans-serif font.

INERTIAL
CONSULTING
SOFTWARE TESTING SOLUTIONS

Brian Kitchener

brian@inertialconsulting.com

48