Optimize Your Tests for Continuous Testing

Software Leadership Academy

1

About Me

- Software Quality Leader since the previous century
- Great companies
  - McDonnell Douglas -> Boeing (F/A-18 & C-17 Avionics)
  - Phoenix Technologies (BIOS)
  - Alcatel Lucent – Network Management for DSL
  - Intuit – Quickbooks, Accounting for Small Business
  - Concur – Travel & Expense Management
  - First Data – POS platform for small merchants
- BS Computer & Electrical Engineering (Purdue)
- MS Computer Science (Washington University)
- MBA (San Jose State University)
- Consulting & Leadership development with Software Leadership Academy (http://swleadership.com)

2

First, A Story

3

The Challenge

- 110 Billion LOC being created every year*
- 1.5 defects/loc is the industry average density+

* Application Security Report 2017

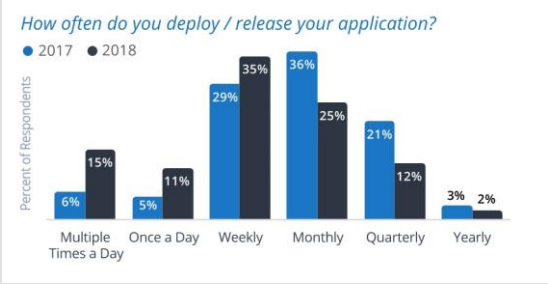
+ Coverity Scan Report 2012

Image: Creative Commons, https://www.flickr.com/photos/tiffanyday/4233056466/

4

Delivery Cadence

- Weekly is the new Monthly



Source: SmartBear State of Testing Survey 2018

5

Opportunity of Continuous Testing

- Supports a Faster Release Cadence
- Exposes Technical Debt
- Reduces Feedback loop
  - Bug creation to bug detection

6

Problems with lots of Automated Tests

- Setup – Deployment/configuration
- Single point of success – White Tower effect
- Product changes – keeping tests accurate
- Triage – turning raw data into results
- Becomes its own project
- Test Environment maintenance
- Test Environment stability
- Test Environment scalability

7

Decomposition Method

- Understand the big problem
- Break it down into a smaller set of problems
- Evaluate each using a model
- Prioritize and Improve

8

Inventory Your Automated Tests

Suite Name		
Suite A		
Suite B		
Suite C		
Suite D		
Suite E		
Suite F		
Performance Test		
Security Suite		

9

Rate By Value

Suite Name	Value	
Suite A	Very High	
Suite B	High	
Suite C	Very High	
Suite D	Moderate	
Suite E	None	
Suite F	Moderate	
Performance Test	Very High	
Security Suite	Very High	

10

Value of Tests

- Simple Rating:
  - Very High
  - High
  - Moderate
  - No Value
- Subjective Opinion
  - Yours or a team discussion

11

Image: Public Domain

12

Value of Tests

- Level of confidence in the suites as indicator of quality
  - How likely would you still release if these tests gave a failing indication?
  - How much effort would you put into getting these tests running if they failed for some reason?
  - Importance of the areas of your product tested?
- Reliability of the tests
  - How often do they provide accurate results?
  - If the tests indicate a failure, does that almost always results in a real product bug?

13

Examples

- Tests a vital part of the app, the main reason people purchase/use your product
- Sometimes, the results are a little flaky and need someone to re-run those tests

High

14

Examples

- Suite has traditionally been run as part of regression tests
- But, no one knows what the results mean
- The original author has moved on
- We run these just because they exist

No Value

15

Rate By Value

Suite Name	Value	
Suite A	Very High	
Suite B	High	
Suite C	Moderate	
Suite D	Moderate	
Suite E	None	
Suite F	Moderate	
Performance Test	High	
Security Suite	Very High	

16

Time to Execute

- Setup + Execution + Reporting

17

Add Execution Time

Suite Name	Value	Time (minutes)
Suite A	Very High	7
Suite B	High	45
Suite C	Moderate	32
Suite D	Moderate	25
Suite E	None	90
Suite F	Moderate	75
Performance Test	High	125
Security Suite	Very High	83

18

Pivot The Table

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

19

High Value / Fast

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

- Run These with the every build
- Your definition of "quick" may vary

20

High Value / Slower

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

- Run These Continuously
- If a new build is available, run the tests as soon as the previous execution completes.

21

High Value / Slow

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

- Run These every day (likely at night)
- Get results every morning

22

Moderate Value

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

- Run These once per release cycle or once per week
- Your times may vary

No Value

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

- When should we run these?
- **Don't run these** – improve the value or drop the tests

Improve The Tests

Make Tests Faster

Add Value

	< 10 minutes	< 1 hour	> 1 hour
Very High	Suite A		Security
High		Suite B	Performance
Moderate		Suite C Suite D	Suite F
No Value			Suite E

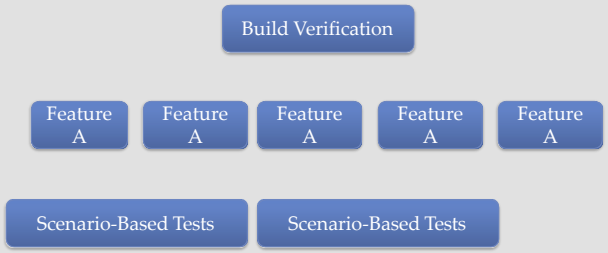
25

5 Tips to Optimize Tests

- Create Tiny, but valuable, test suites
- Refactor the test setup
- Be smart with your wait times
- Trigger tests automatically
- Run tests in parallel

26

Create Tiny Test Suites



27

Refactor Test Setup

- Test Data Preparation
  - Use API instead of UI
- Prep Systems "off line"
  - Containers
  - VMs

28

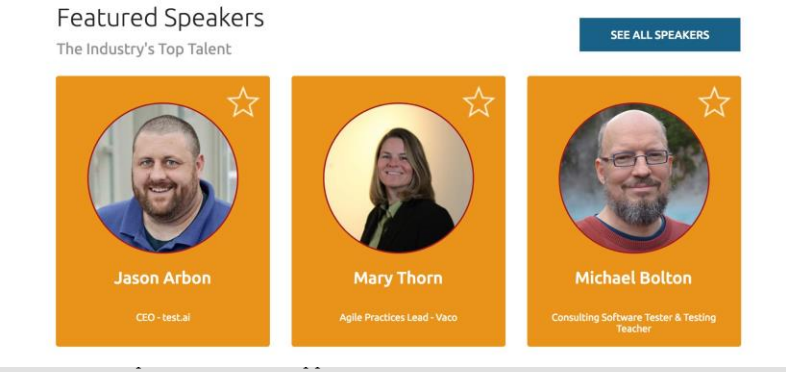
Be Smart with Wait Times

~~time.sleep(5)~~
~~time.sleep(10)~~
~~time.sleep(15)~~
time.sleep(30)

Use "wait" instead

Waits for an event to happen – can proceed once the condition has been fulfilled.

29




```
speakers= WebDriverWait(driver, 15).until(ec.visibility_of_element_located(
(By.XPATH, "//*[@id="tile_speakers"]/a")));

speakers.click()
```


30

Trigger Tests Automatically

- Continuous Integration System
- Source Control
- Deployability
- Results Reporting
- Logging / Telemetry

31

Run Tests in Parallel

- Multiple machines / test clients
- VM / Containers
- Cloud Providers (check out the STPCon sponsors)

32

Questions

- Connect with me on LinkedIn
  - https://www.linkedin.com/in/ruberto/
- Follow me on Twitter
  - @JohnRuberto
- Read my Blog:
  - Blog.ruberto.com

33

Case Study

- Test Suite: 500 UI driven tests across the entire product
- High confidence in the tester and coverage
- Passing this suite was considered mandatory
- But…
  - Single Point of Success
  - Tests should have taken 5 hours, took 5 days
  - Lots of test / tweak / re-test iterations

34

Agenda

- Automation Overview
  - Suites, System Test,
  - Pyramid
  - Regression
  - Focused area
- Problem statement (automated tests take a lot of manual effort), Opportunity (Continuous Testing)
- Inventory and Evaluate: Value + Time
- Value: No Value, Moderate, High, Very High (examples)
- Time: Minutes, include setup & trigger effort
- Chart (Shift Left, Shift Up)
- 1. Tiny, but valuable suites. (DBT, Feature Sanity, etc.)
- 2. Refactor Test Setup
- 3. Be smart with waiting
- 4. Trigger tests automatically
- 5. Run tests in parallel

35