SMARTBEAR

Accelerating Your Test Execution Pipeline

Bria Grangard

Who Am I?

- SmartBearSoftware
  - Automated UI functional testing tools & test management solutions
- Stay in Touch!
  - @Bria_Grangard
- Education
  - Dartmouth: AB in Engineering, BE in Biomedical Engineering, MEM with a healthcare focus
- What do I love to do?
  - Run, dance, play board games (Settlers of Catan anyone?)

Today's Agenda

- SmartBear Overview
- Testing Overview
- The Importance of a Test Framework
- What's the Secret Sauce to Automation
- Parallel Testing Overview
- Benefits Of Parallel Testing
- Q & A

We provide tools for development, testing, and operations teams

SMARTBEAR

to create great software, faster than ever.TestComplete

SoapUI Pro

6.5M+

Users

Countries**194**

Companies**22K+**SwaggerHub

CrossBrowserTesting

- HQ in Boston, MA, USA, with **7 offices** globally
- Founded in 2009

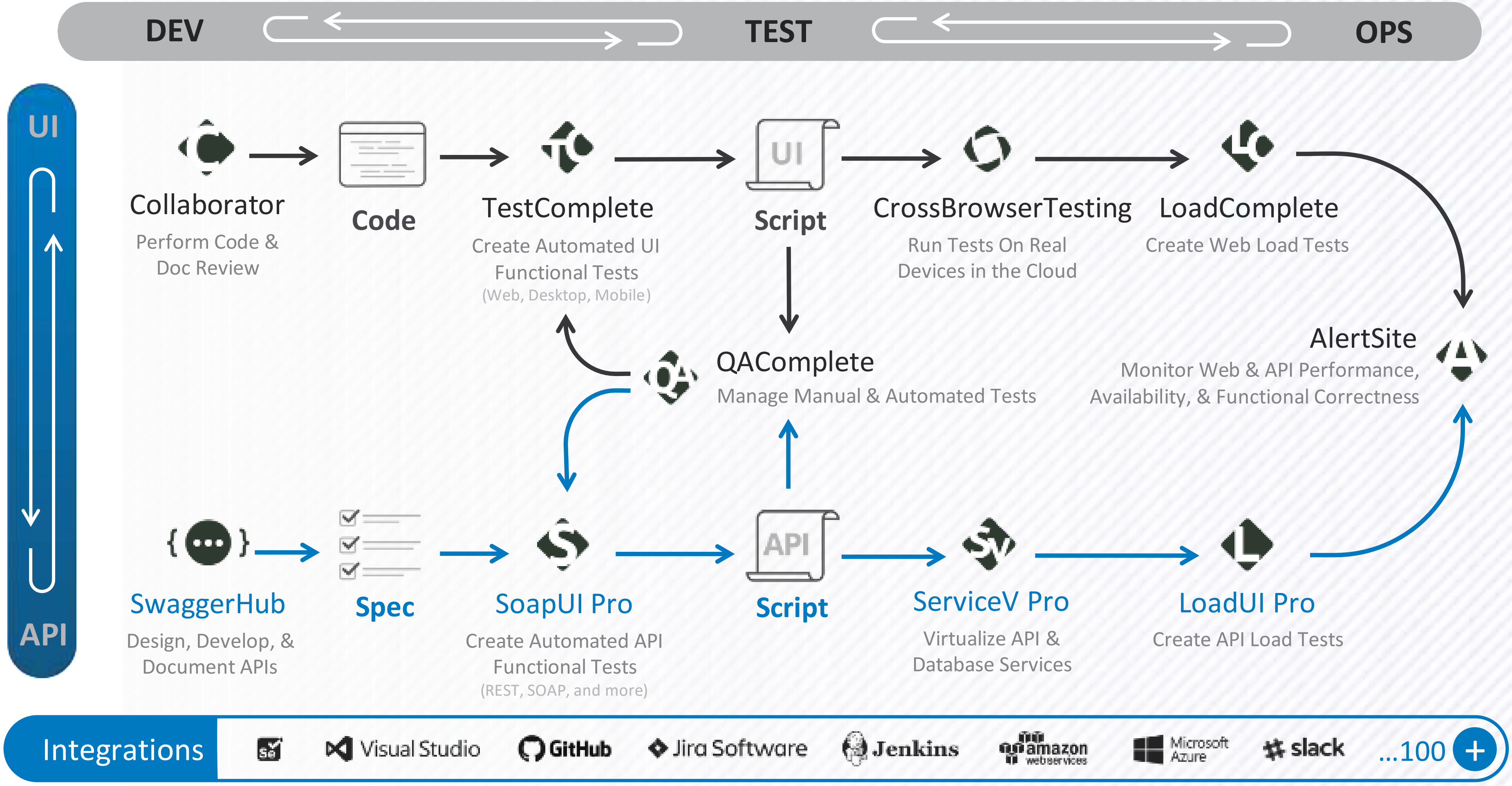
QAComplete

- Open Source **Innovator** (Swagger & SoapUI)

AlertSite

Accelerate SDLC Workflows |Improve Quality at Every Stage |Realize Rapid Time-to-Value

Create Great Software, Without Tradeoffs



DEV TEST OPS

UI API

Collaborator
Perform Code & Doc Review

Code

TestComplete
Create Automated UI Functional Tests
(Web, Desktop, Mobile)

Script

CrossBrowserTesting
Run Tests On Real Devices in the Cloud

LoadComplete
Create Web Load Tests

QAComplete
Manage Manual & Automated Tests

AlertSite
Monitor Web & API Performance, Availability, & Functional Correctness

SwaggerHub
Design, Develop, & Document APIs

Spec

SoapUI Pro
Create Automated API Functional Tests
(REST, SOAP, and more)

Script

ServiceV Pro
Virtualize API & Database Services

LoadUI Pro
Create API Load Tests

Integrations

BDD

Machine Learning

Agile

What’s Going on in the Testing World?

DevOps

AI

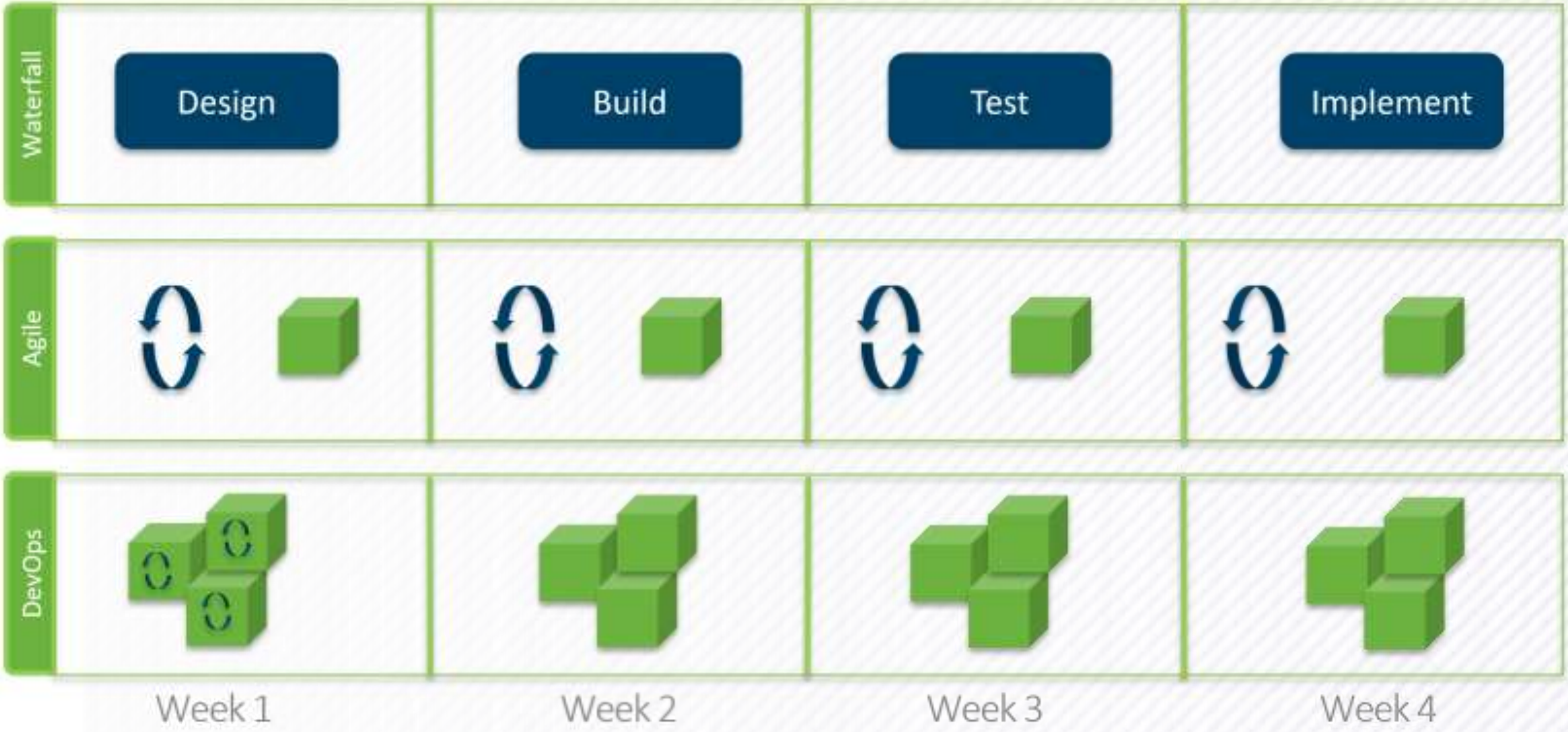
Shift Left

Automation

There are bottlenecks in today's development processes.

- Iron triangle trade-off
    - Teams today are constantly feeling pressure to deliver software faster, without compromising quality
- Automation might ramp up
    - There is only a certain point as to how scalable automation can be
- Test environments are often the root-cause of the bottlenecks
    - They are very time consuming and costly

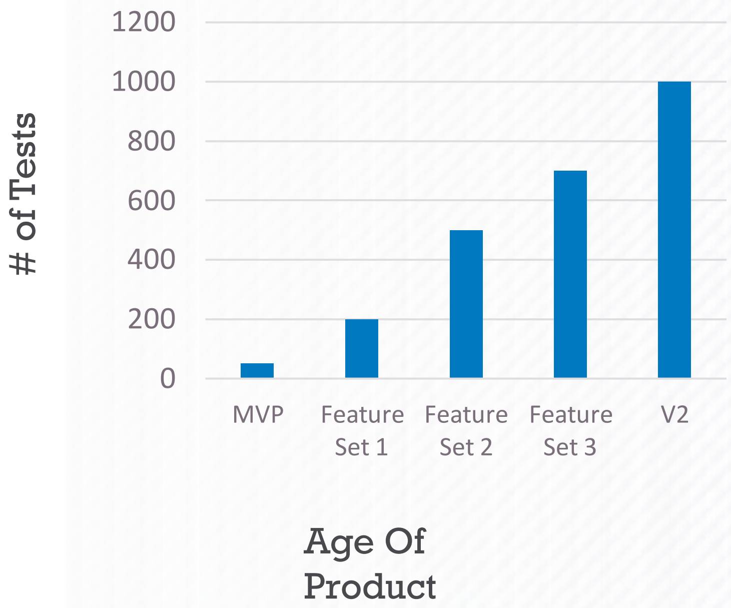
The promise of the new software delivery cycle

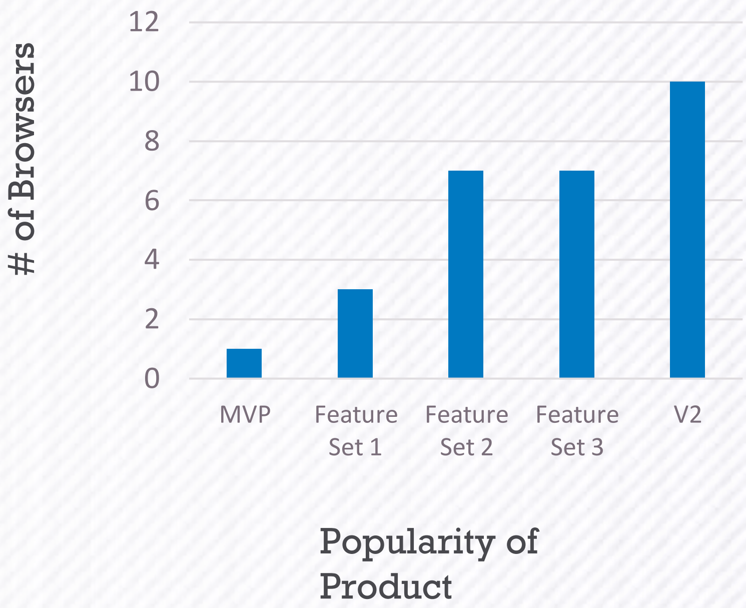


	Week 1	Week 2	Week 3	Week 4
Waterfall	Design	Build	Test	Implement
Agile				
DevOps				

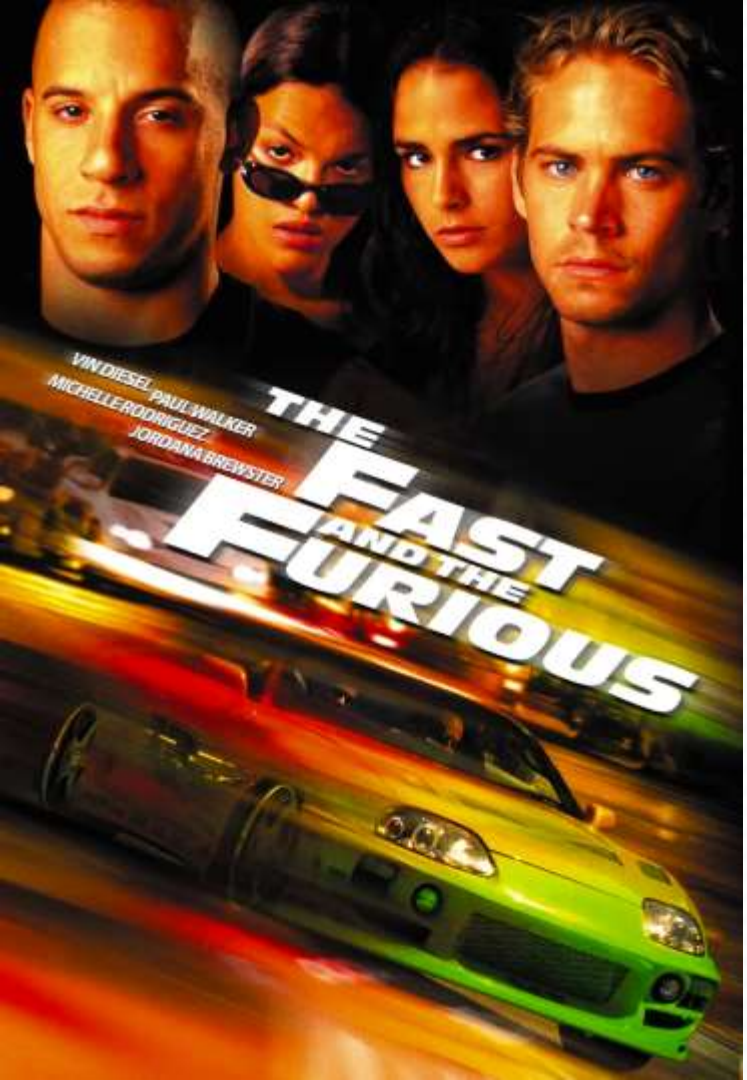
Time Consuming Nature Of Web Testing

More Features = More Testing





1. ~~Test Less~~ More Bugs
2. ~~Hire More Testers~~ Increased Cost
3. Diversify Your Testing
4.

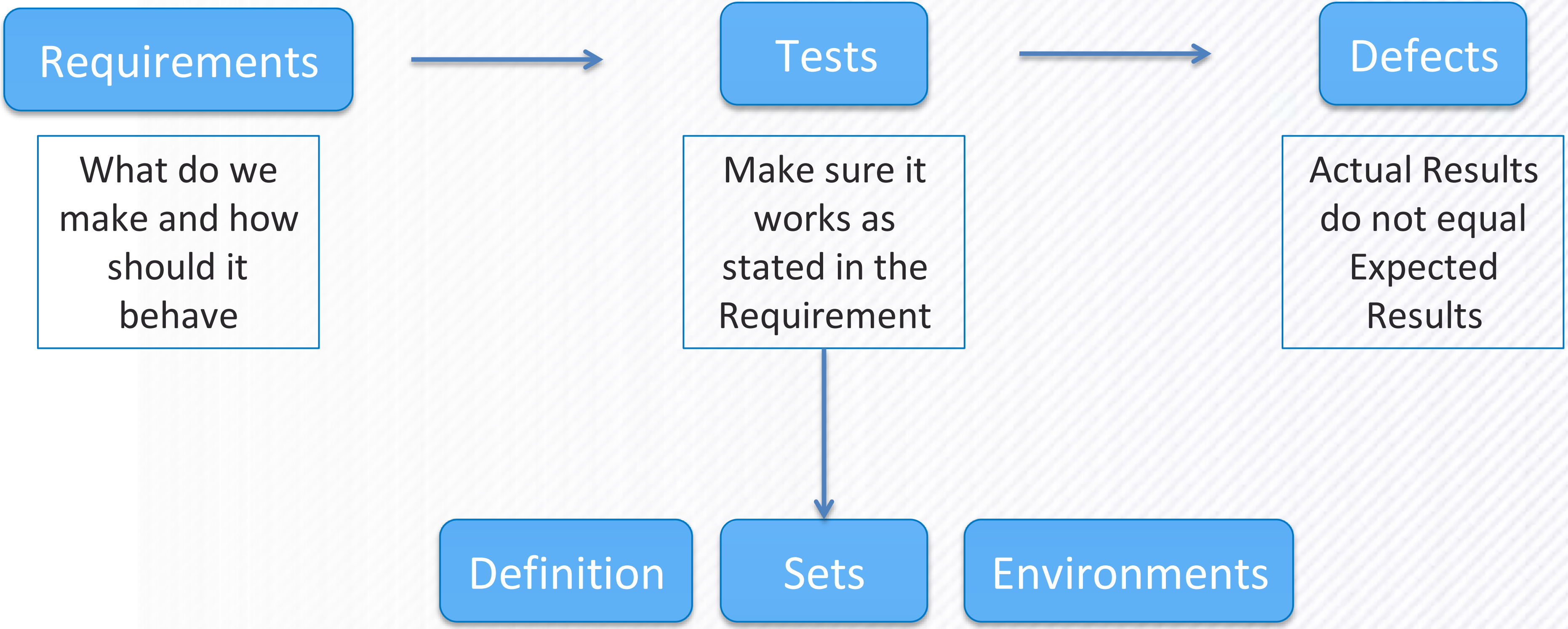


Now… How Can We Go Faster?

1. Test Frameworks
2. Automation
3. Parallel Testing

Level 1: Test Frameworks

The Basics of a Test Framework



Requirements

What do we make and how should it behave

Tests

Make sure it works as stated in the Requirement

Defects

Actual Results do not equal Expected Results

Definition

Sets

Environments

What is a Test Framework?

A Test Framework:

Links tests to other SDLC items

Is NOT a Test Automation Framework but often contains one

Allows for rapid creation of tests from reusable components

Separates data from logic (REUSABILITY)

Provides a standardized test “language” and reporting structure for an applicatio under test

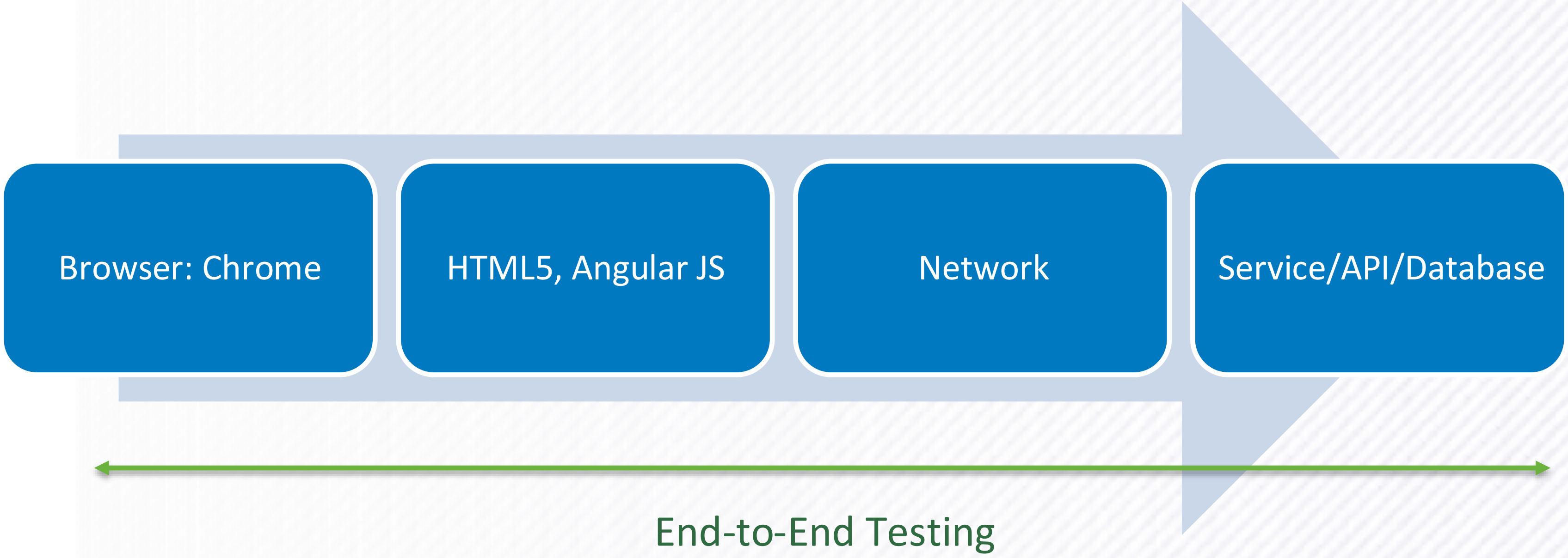
Elements of a Test Framework

- Library: A repository of all your decomposed scripts, separated into their components
- Test Data Sources: A repository of all data sources
- Helper Functions: A repository of all decomposed test scripts, automated or manual, that are inputs or checks

- Test Environments: A list of all covered testing environments, broken out by type (OS, browsers…)
- Modules: The combination of library items with any helper functions and test data sources–plus environments
- Structure / Hierarchies: The “folder” structure of modules

Level 2: Figure Out What Tests Should be Automated

There are many types of testing that need to be done…



Browser: Chrome

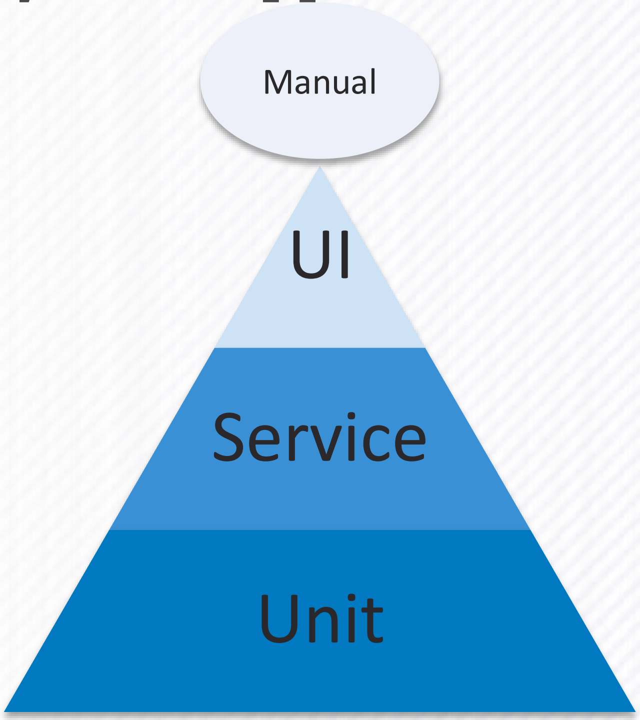
HTML5, Angular JS

Network

Service/API/Database

End-to-End Testing

Automated Testing Pyramid Approach



A Little Manual v Automated Math

Product v2

# of Test Cases	1,000
# of Browsers Supported	10
Total Test Cases	**10,000**
Avg Test Run Time	4 Min
Total Test Time	**666Hrs**

With 5 manual testers, that is 3.5 weeks of testing

Product v2

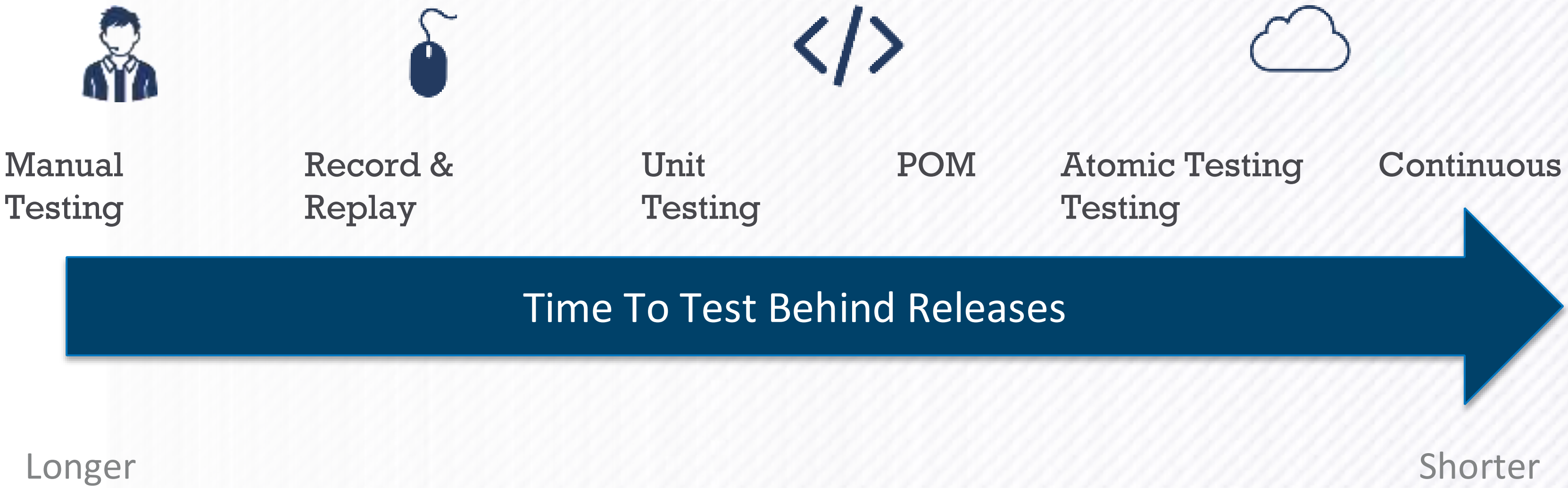
# of Test Cases	1,000
# of Browsers Supported	10
Total Test Cases	**10,000**
Avg Test Run Time	.5 Min
Total Test Time	**83 Hrs**

With 2 QA engineers, That is 1 week of testing

Decide on What to Automate

- Environment Setup/Teardown
- Data Entry
  - Form Filling
  - Varying data inputs in a repetitive process
- Exposing backend data (APIs, DB table, etc.)
- Repetitive/boring tasks that are prone to inattention errors
- Tasks with high reuse value across many workflows
- Tests with timing or screen responsiveness as a criteria for success
- Many non-functional test types, such as performance testing
- Capturing Results

Speeding Up Your Pipeline



Manual Testing

Record & Replay

Unit Testing

POM

Atomic Testing

Continuous Testing

Time To Test Behind Releases

Longer

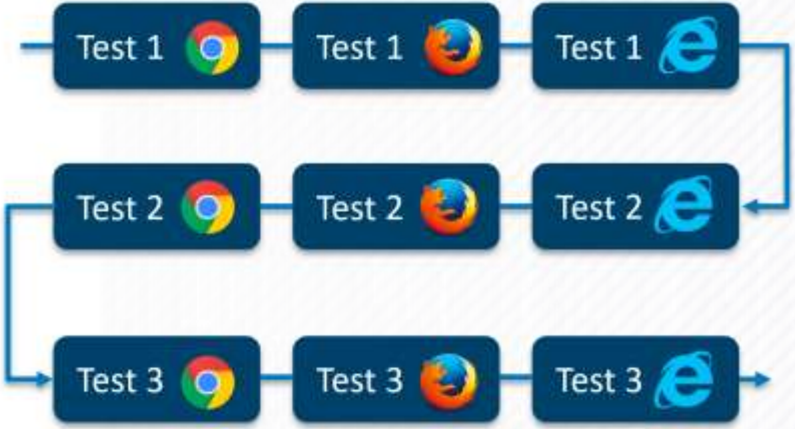
Shorter

But my Dev team says I have days to test, not weeks …

Level 3: Go Faster.

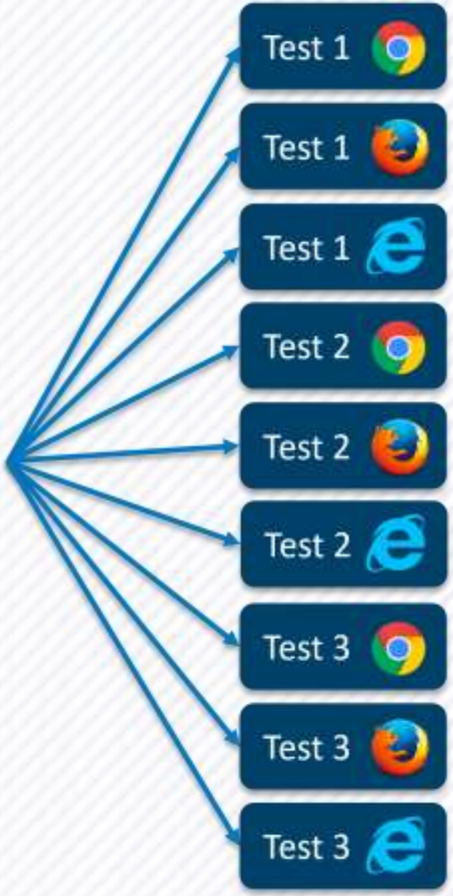
Let's Go Faster!

We can, with parallel testing



Running tests sequentially, we were able to run our tests in 1 week

With 20 Parallel executions, we can run our entire test suite in only 4 hours



Types of test to run in parallel

Most effective tests to see ROI from for parallel testing

Cross Browser Testing

Testing across different browsers and devices is one of the most time consuming aspects of testing the front end of your website or web application. Run more tests, against more browser configurations by running them in parallel.

Unit Testing

According to the testing pyramid, Unit tests should be your most abundant test type in your entire testing suite. Because of this, running 14,000 unit tests in under an hour is really only possible with a massive parallel testing infrastructure investment.

Regression Testing

Because deployments are happening at such a rapid pace, regression testing is on of the best ways to have a type of “testing version control” making sure the functionality of the new build, matches that of the last stable build. Running these tests in parallel allows more to be tested.

Smoke Testing

Need to get your minimum testing done in the next 20 minutes while you push a hot fix? Only way to do that is to run them in parallel, allowing you to get the most testing done in the shortest amount of time.

Benefits of Parallel Testin g

1. Quick Deployments
2. Faster Feedback
3. Cross Browser Testing
4. Better Test coverage
5. Saves Valuable Time

Key Takeaways

1. Test frameworks are important and a key to success.
2. When automating—choose which tests to automate wisely
3. Parallel testing let’s you go faster.

Questions?

Thank You!