OUTLEARN

TESTING JAVASCRIPT APPLICATIONS

DANIEL JOHNSON

WHO AM I?

Software Architect at Outlearn

Instructor at General Assembly

JS, Ruby, Python, PHP, Databases

Backbone, Angular, Ember, Rails, Django

BS from Northeastern University

Can be found skiing or mountain biking

linkedin.com/in/balancerockmedia

WHY TALK ABOUT TESTING?

You write software that needs to work

Code it like there will be a larger team tomorrow

QA teams add a lot of value, but it's not to manually test all your code on every deploy

It will make you a better programmer

You never want to be the developer in the demo who's feature doesn't work

IMAGINE THAT…

- You have 12 person distributed development team
- You have a 5 person distributed QA team
- Building a large scale application on a tight deadline
- The testing plan didn't quite make it into the budget
- Incomplete unit test coverage
- No functional tests
- One manual regression cycle by QA team takes 5 days
- You end up in a never-ending regression cycle with no way out

IMAGINE THAT…

- You have 2 person development team
- You have no QA team
- Building a small/medium scale application on a tight deadline
- You figure you'll write tests as needed
- Client requests a new feature that requires you to do a significant refactor
- You get the new feature to work, but break half of the other features in the process

START THE CONVERSATION!

Testing is a tough sell, but it's WORTH IT

You boss might not get it, clients probably won't get it, your friends won't be impressed, but it's WORTH IT

Everybody on your team will benefit (not just the developers), and it's actually kinda fun!

WHY?

- **Devs** - can own and prove the quality of their work
- **QA** - get to focus on the real manual QA work and push their career forwards by learning to write more programatic tests
- **PM/leadership** - less stress (entire team is happier)
- **Client** - just wants their stuff to work, when it does, $$

BACK TO BASICS

Unit Tests

- Test a small single unit of code
- Run in isolation
    - If other pieces are needed they are mocked
- Used for code coverage metrics

Integration Tests

- Test that multiple units of a system work together
- Can be anything from a couple functions, to something that requires a resource like a database, to a full system wide test

BACK TO BASICS

Functional and Acceptance Tests

- Used by the development and/or QA team to check that features of the application match up to the spec
- Often match up to agile stories
- Functional tests run using a headless browser by dev or QA team
- Acceptance tests run in a production like environment with real browsers by the QA team or client

DECIDING ON AN APPROACH FOR JS…

Testing is made difficult by the fact that JS is very often tightly coupled to HTML/CSS

Unit tests are great, code coverage metrics for JS aren't!

Integration tests are also great, but at what cost?

So how do you choose?

WRITE UNIT TESTS FOR…

Business logic (i.e. validation, math, algorithms, etc)

The M in MV*

Be cautious of any unit test that requires mocking the DOM

Be cautious of unit tests that don't actually test anything and artificially bump line coverage

WRITE INTEGRATION TESTS FOR…

Critical features and paths through your app (i.e. login)

The V and * in MV*

These most commonly fall into the functional and acceptance testing category

It's too expensive and not feasible to test every single user flow

DON'T FORGET THE SERVER SIDE TESTS!

UI tests are great, but you still need server side testing (both unit and integration)

TESTING ANTI-PATTERNS

- **Second Class Citizens** - test code isn't as well refactored as production code, containing a lot of duplicated code, making it hard to maintain tests.
- **The Local Hero** - A test case that is dependent on something specific to the development environment it was written on in order to run. The result is the test passes on development boxes, but fails when someone attempts to run it elsewhere.
- **Chain Gang** - A couple of tests that must run in a certain order, i.e. one test changes the global state of the system (global variables, data in the database) and the next test(s) depends on it.
- **The Dead Tree** - A test which where a stub was created, but the test wasn't actually written.
- **The Liar** - An entire unit test that passes all of the test cases it has and appears valid, but upon closer inspection it is discovered that it doesn’t really test the intended target at all.

AUTOMATE EVERYTHING (CONTINUOUSLY)!

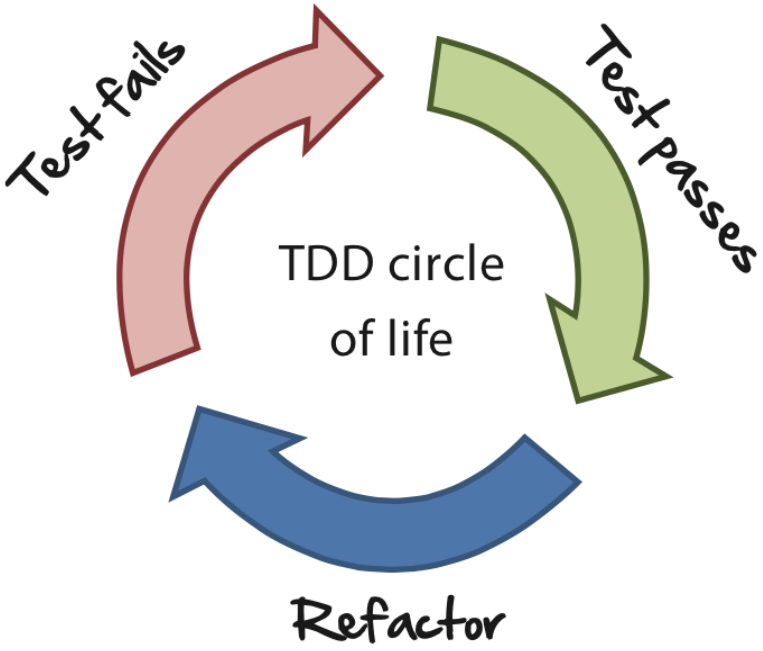
Unit and integration tests should be hooked into your CI/CD tool of choice and should be run at regular intervals as part of your dev workflow

https://semaphoreci.com

https://codeship.com

IS THIS AGILE? IS ANYTHING?

If agile is all about delivering working software and customer value, then yes, testing is crucial



THE REALITY IS…

True TDD/BDD is hard to achieve in UI development and while not impossible, I don't believe in that as the top priority

The goal is to have well tested features, if the tests get written in the middle of development or even at the end, but they are good tests and force some refactoring, at the end of the day the goal was still achieved

STEPS OF DONENESS

- Visual design
- Technical design API
- Technical design UI
- API doc for JSON contract
- **Implementation API**
- **Implementation UI JS**
- Implementation UI HTML/CSS
- Browser/platform test
- **Code review**
- Visual review
- Business/client review

BEHAVIOR DRIVEN DEVELOPMENT (BDD)

BDD is a way to get an entire team using a single format to describe the system

The entire team should be invested in testing

Cucumber.js makes this possible and can fit in with most of the other libraries and tools

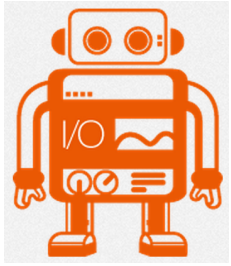
FEATURES AND SCENARIOS

Feature: Shopper can add an item to their Grocery List
As a grocery shopper
I want to add an item to my grocery list
So that I can remember to buy that item at the grocery store

Scenario: Item added to grocery list
Given I have an empty grocery list
When I add an item to the list
Then The grocery list contains a single item

Scenario: Item accessible from grocery list
Given I have an empty grocery list
When I add an item to the list
Then I can access that item from the grocery list

TOOLS AND LIBRARIES



WHAT DO YOU NEED?

Assertion library (i.e. node, chai, etc)
http://chaijs.com

A way to do spies, stubs and mocks
http://sinonjs.org

Test framework to run the assertions, organize test suites and modules and handle async code (i.e. mocha, jasmine, cucumber, etc)
http://mochajs.org

A test runner (i.e. karma, intern, etc)
http://karma-runner.github.io

Browser automation (selenium, webdriver.io, etc)
http://webdriver.io

HOW TO CHOOSE?

No framework? No worries, use whatever combination you want!

Backbone - not opinionated

React - Jest

Angular - Jasmine, Karma, Protractor

Ember - Ember CLI, QUnit

TIME TO WRITE SOME CODE!

Mocha unit test walkthrough
Mocha live coding exercise
Jasmine unit test walkthrough
Jasmine live coding exercise

Rock paper scissors walkthrough
Rock paper scissors live coding exercise

Webdriver.io functional test walkthrough

Angular testing walkthrough

THE END

Questions?


OUTLEARN