W6

Performance Testing
Wednesday, October 23rd, 2019 11:30 AM

Automating the New York Times Crossword

Presented by:

Phil Wells

New York Times

Brought to you by:

888-268-8770 ⋅ 904-278-0524 - info@techwell.com - http://www.starcanada.techwell.com/

Phil Wells

Phil Wells has been a software quality practitioner for over a decade. Now, Phil is a software engineer in test for the New York Times games team. This team maintains the most popular crossword product in the world. Phil works to ensure that this team builds quality into every new feature and game they deliver. Phil likes to go beyond writing tests and building infrastructure for delivery. He also acts as a coach for his peers in web development, teaching and advocating for modern test practices and technologies. People have all sorts of funny ideas about what Phil does every day. Phil does not construct the puzzle content for the crosswords. Phil does not program an AI to solve crosswords, although that would be awesome. Phil does not know Will Shortz. If you see Phil walking around the conference, feel free to say, "Hi, Phil!"

Automating the Crossword

Testing Web Games at the New York Times

Phil Wells
@thephilwells
phil.wells@nytimes.com
October 2019

Games at the New York Times

- Team composition and processes
  - Cross-functional team building games for web, iOS, and Android
  - Agile teams combining test automation with manual QA processes
- What we've been up to
  - Supported and updated the online NYT Crossword
  - Released new games: Spelling Bee! Letter Boxed!
- Big theme for crossword improvement: Removing Shame for solvers

The Puzzles Get Tough

Doing Monday's Puzzle

Doing Saturday's Puzzle

@thephilwells

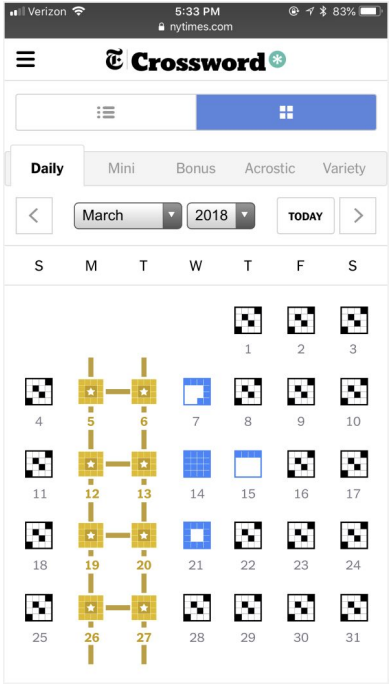
Removing shame for puzzle solvers

- “Shame” is a big word
- Make solving enjoyable for Monday/Tuesday solvers
- Less Shame, More Game!

How We Helped Solvers

Milestone Animations

Vertical Streaks



Autocheck

Software development is a puzzle, too

- A good outcome is possible
- The path is not straightforward
- We lose ourselves in the work
- It’s “our puzzle”
- When we can’t get it right, we might feel ashamed

Removing Shame In Software Development

- Tests are there to remove shame from software development
- Releasing code with bugs - ***SHAME!***
- Tests that do more harm than good - ***SHAME!***

Who you gonna call?

Shame Busters!

Responsibility

Running Often

Really Quite Fast

Reliability

@thephilwells

Responsibility

Who owns quality on this team?

Without Shared Responsibility

- “I can’t read the test code.”
- “They can’t read my test code!”
- “I don’t even know what’s covered by our end-to-end tests.”
- “The tests are broken and the test engineer is on vacation. Guess it’ll just stay broken forever.”

SHAME

responsibility

“Web Developers Couldn’t Possibly Create Tests”

Who Needs To Read Test Code?

- Not just a Software Engineer in Test
- Not just the engineer who wrote it
- ***Every engineer on the team needs to be able to read the automated test code***

responsibility

You Can't Test Everything End-to-End

- "Which Tests Should We Automate" - Angie Jones, SauceCon 2018
- Too much maintenance
- Too much time
- Too much noise
- Too much ***code!***

On This Team, Everyone Writes Tests!

- Prioritized test coverage is part of our definition of done
- Features ship with unit and component tests
- Coverage gaps are handled in end-to-end tests using webdriver.io

Feature Milestones

- Test Strategy Kickoff
- Test Coverage Audit Before Launch
- Test Tech Debt Stories During “Code Hardening”

responsibility

When Anyone Can Write A Test...

- Anyone can triage or fix a broken test
- Anyone can see what components the tests are checking
- Anyone can tell new team members how to do it

… anyone can read the tests!

BUSTED!

Run Often

And Report the Results

Infrequent, Unmeasured Tests

- “Whoa, the last time we ran tests on Staging was two months ago. There have been three releases in that time!”
- “I feel like there’s always a sprinkling of flakiness when the tests run, but I can’t pinpoint which tests might fail.”
- “My boss wants to know how much our test coverage has improved this quarter. What info is lost to time? What am I, an archeologist?”

SHAME

run often

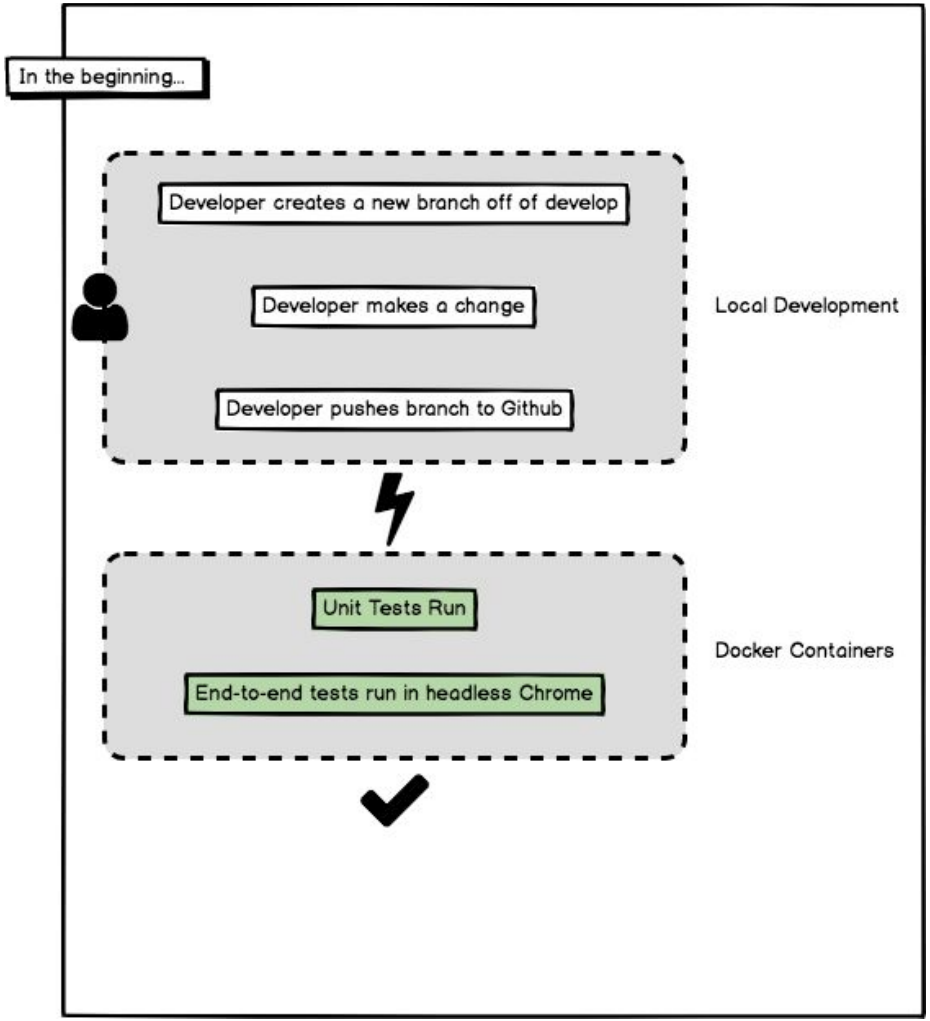
Automate Your Automation

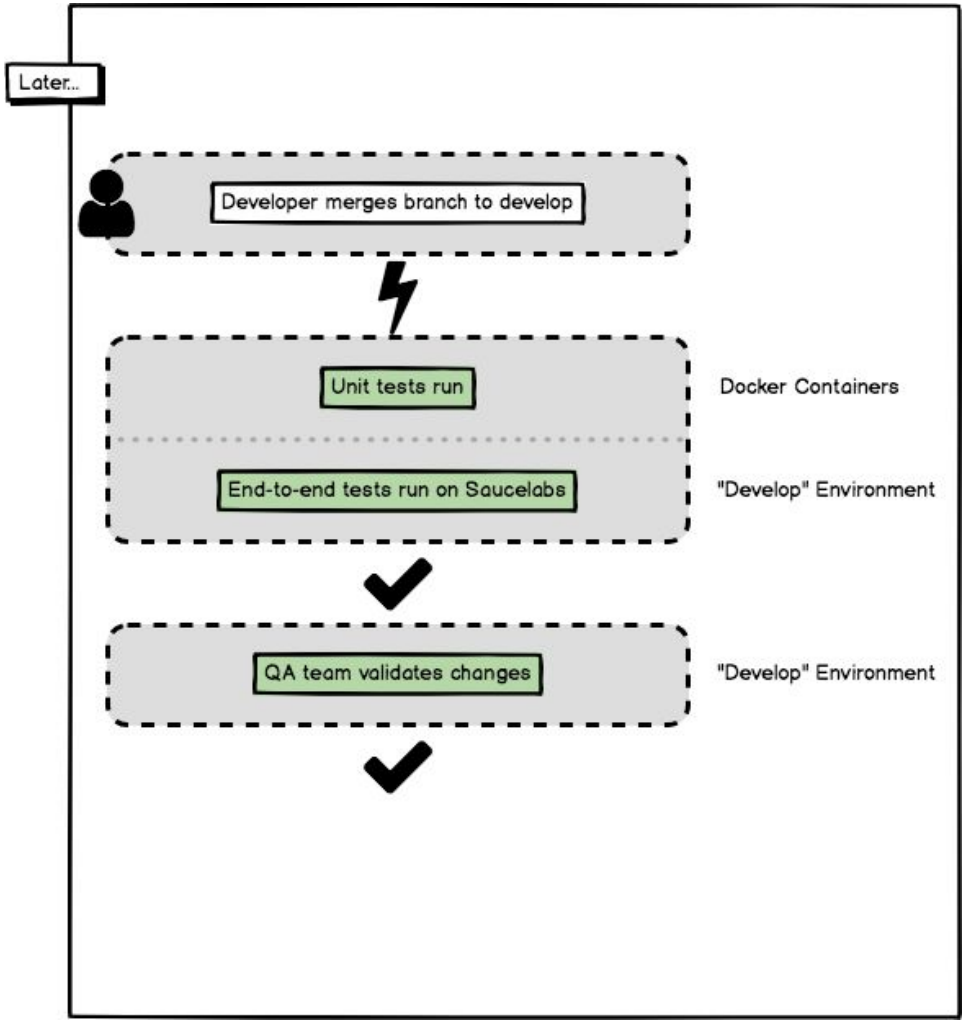
- Continuous Integration
- Sauce Labs
- Github
- Report Portal integration

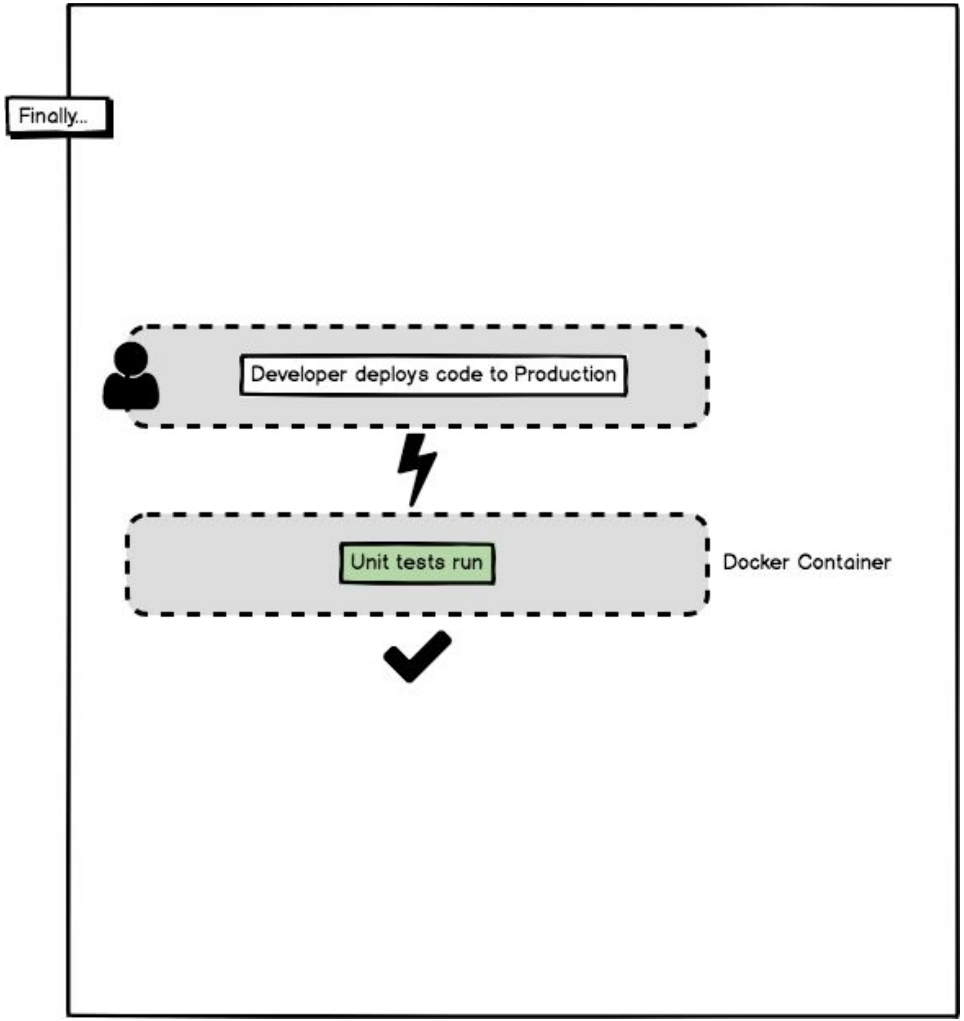
run often

How Often Do We Test?

run often


In the beginning...
Developer creates a new branch off of develop
Developer makes a change
Developer pushes branch to Github
Local Development
Unit Tests Run
End-to-end tests run in headless Chrome
Docker Containers

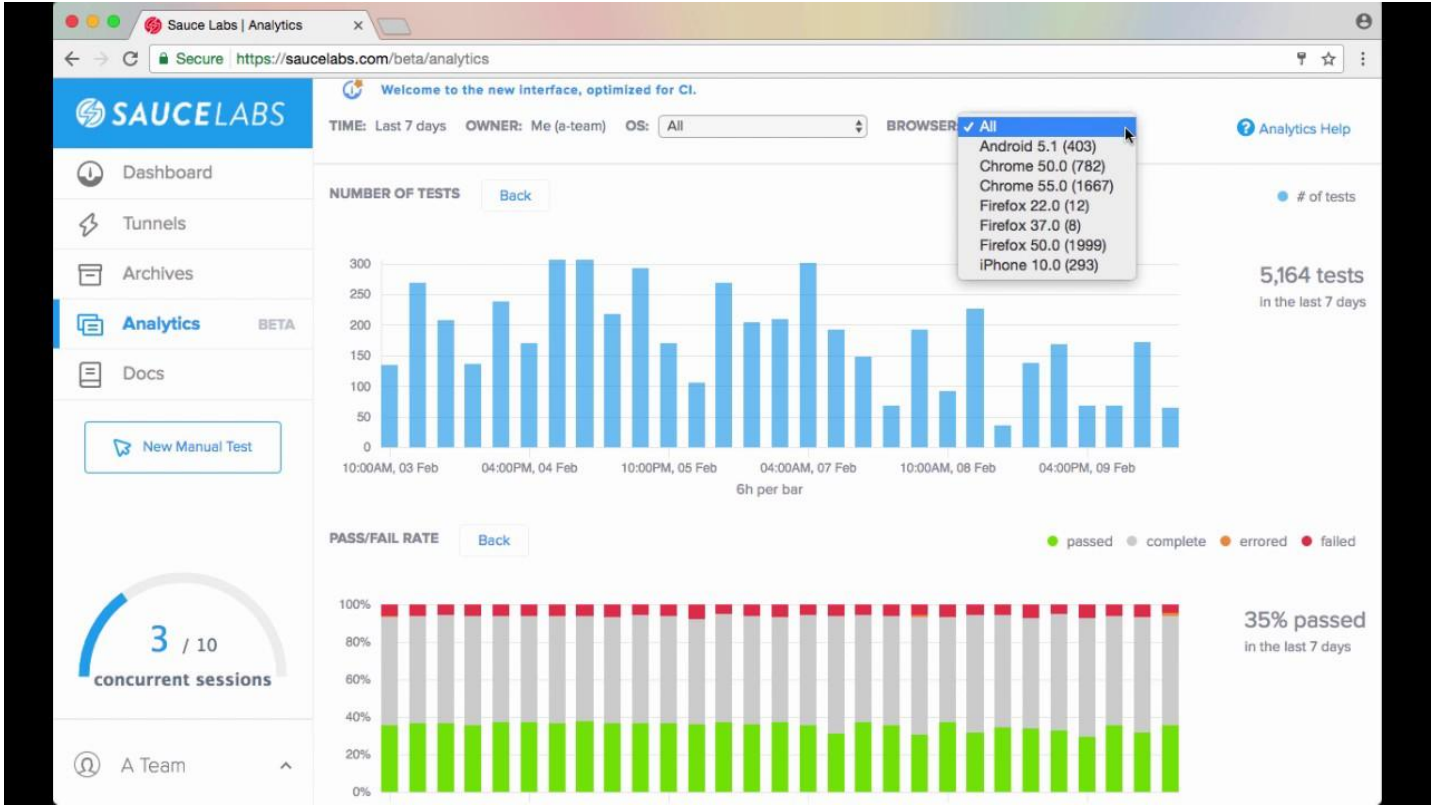

Later...
Developer merges branch to develop
Unit tests run
Docker Containers
End-to-end tests run on Saucelabs
"Develop" Environment
QA team validates changes
"Develop" Environment


Finally...
Developer deploys code to Production
Unit tests run
Docker Container

Pretty Often!

run often

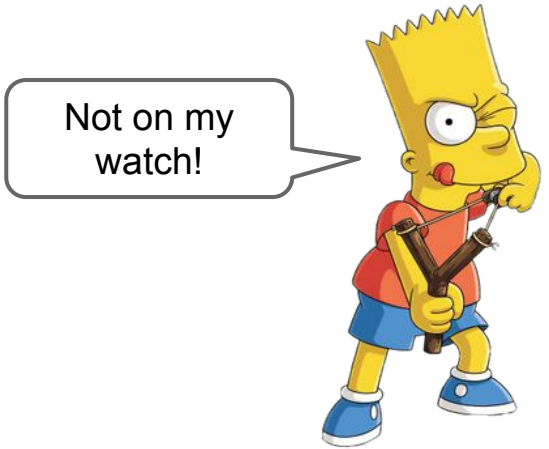
Sauce Labs Analytics



BUSTED!

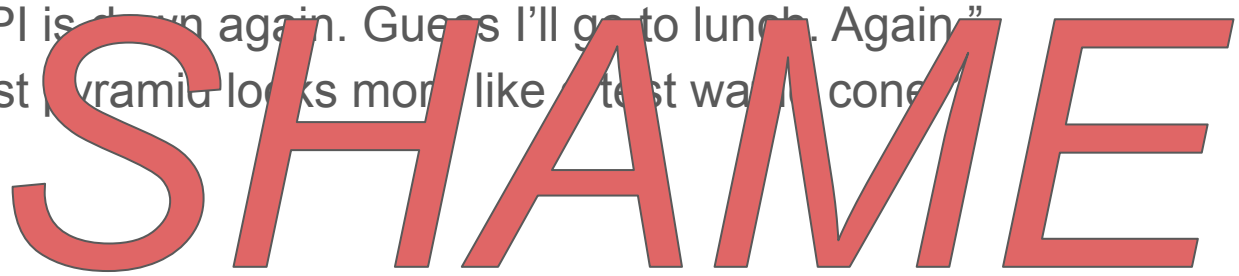
Reliability

Flakiness? Low fidelity?



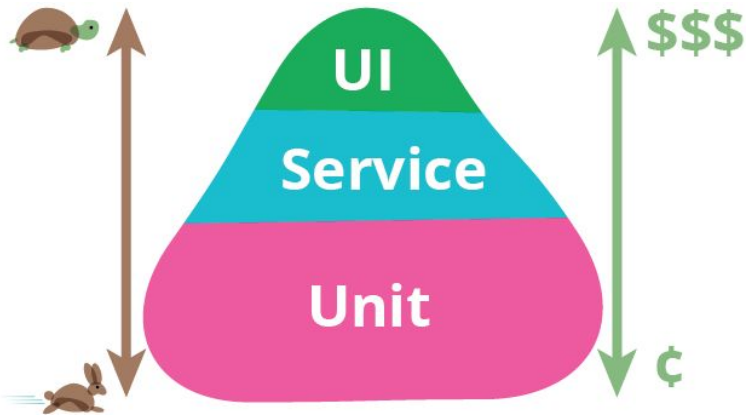
Unreliable Tests

- “Someone tainted these test accounts. Now I have to manually create a new group of test users.”
- “The API is down again. Guess I’ll go to lunch. Again.”
- “Our test pyramid looks more like a test waffle cone.”



Pushing Tests Down the Pyramid

- Couple UI testing with thorough unit test and service test coverage
- Focused UI tests
- Regularly de-dupe tests



Martin Fowler

Getting Rid Of Redundant Tests

- Bugs caught by UI tests should trigger the creation of unit and service tests
- Does this UI test provide any extra value?
- Don’t fall in love with your UI tests!
- “The Practical Test Pyramid” by Ham Vocke
  - https://martinfowler.com/articles/practical-test-pyramid.html

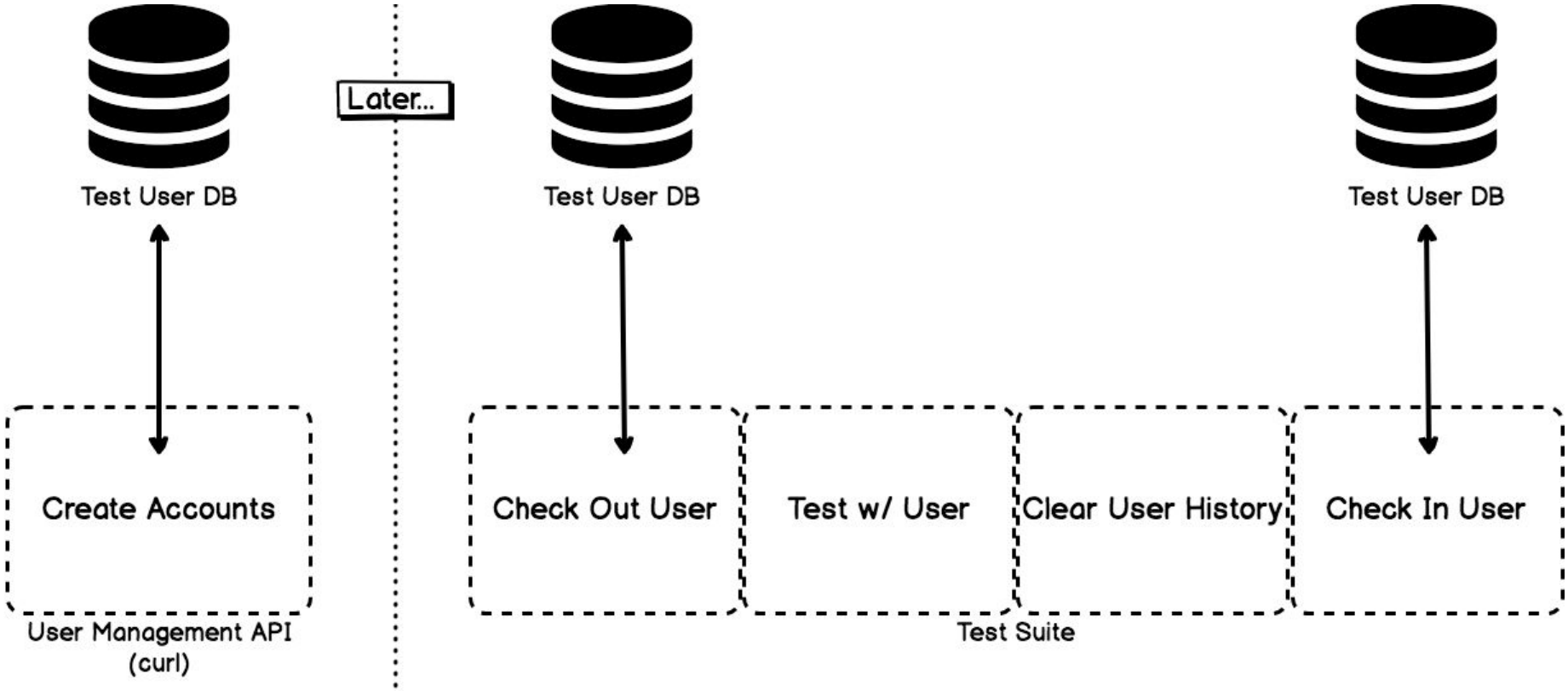
The right tools for the job!

@thephilwells

User Management

User Management

- NY Times User Management Tool
- Easily Generate New Users
- Do Not Share Test Users
- Keep History Clean



@thephilwells

Handling Race Conditions

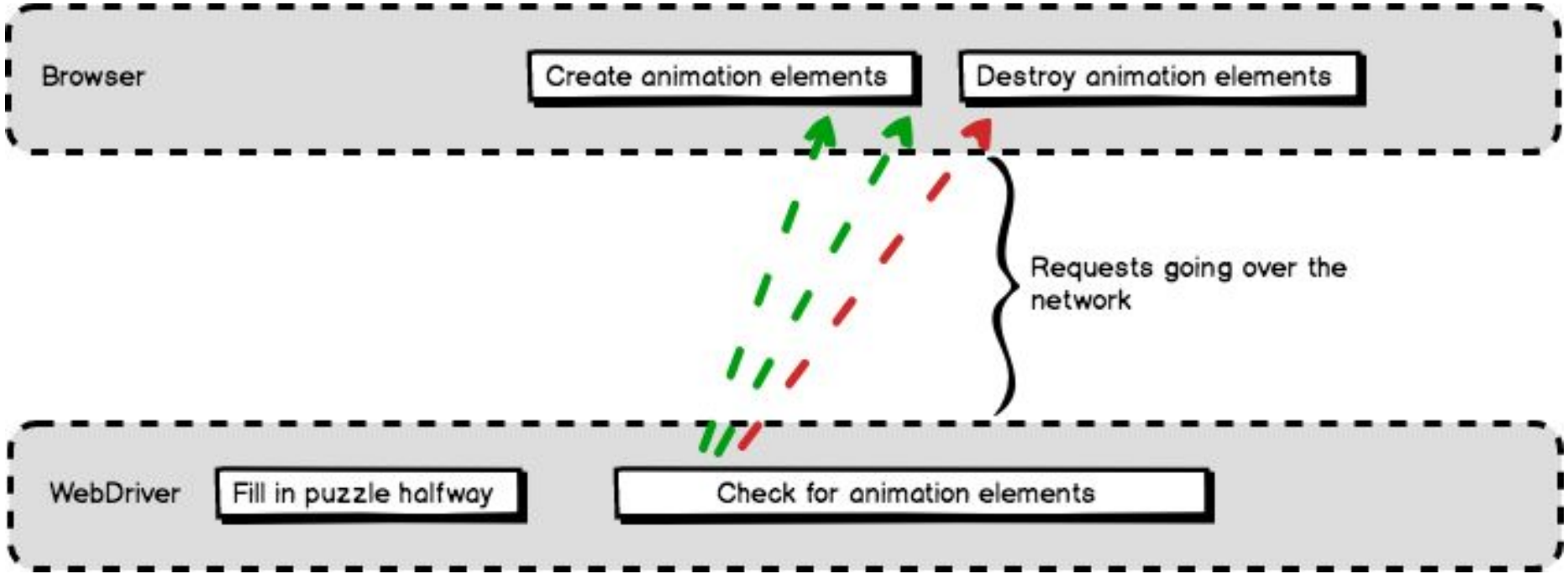
Off to the races…

Problem: Trying to assert existence of short-lived elements on the page.

Case in point: milestone animations!

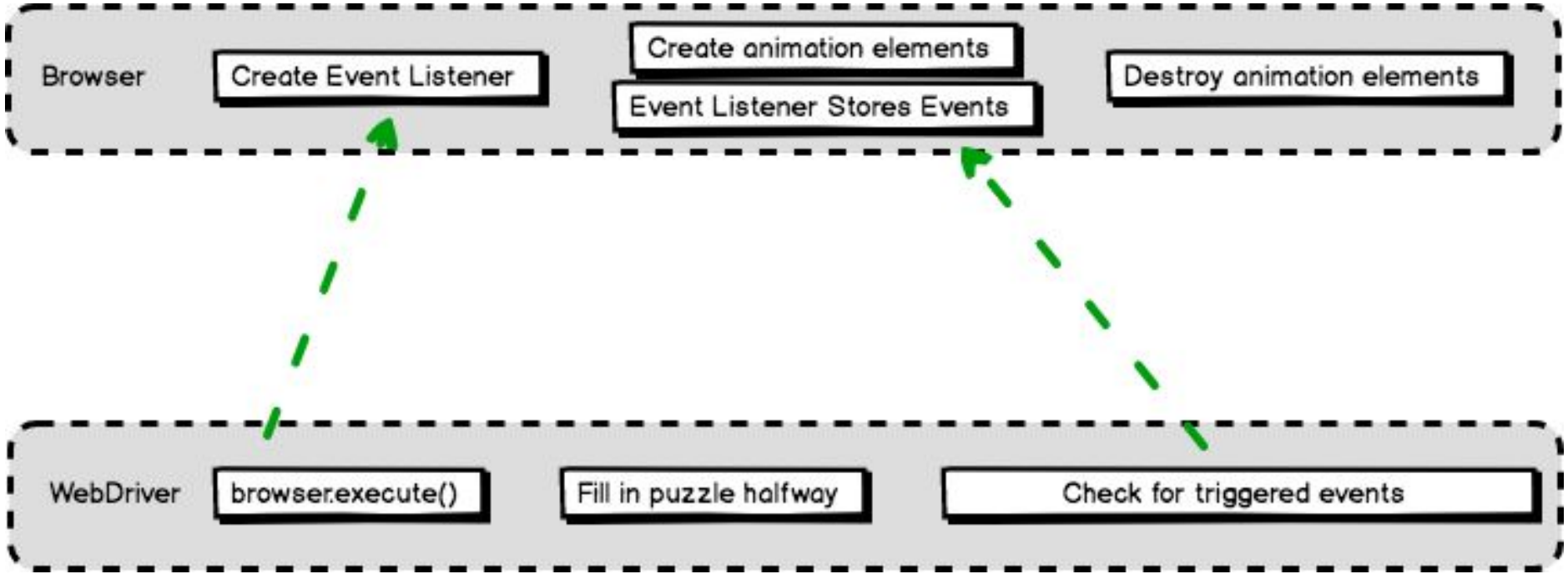
reliability

What Could Go Wrong?



@thephilwells

Record Those DOM Events!



@thephilwells

WARNING:

Here comes the code!

```
1  browser.execute(function (selector, eventName) {
2      window.trackedEvents = window.trackedEvents || {};
3      window.trackedEvents[eventName] = window.trackedEvents[eventName] || [];
4      let targetNode = document.querySelector(selector);
5      let config = {
6          attributes: true,
7          childList: true,
8          subtree: true,
9      };
10     let callback = function (mutationsList, observer) {
11         for (let mutation of mutationsList) {
12             if (mutation.type === 'childList') {
13                 window.trackedEvents[eventName].push(mutation);
14             } else if (mutation.type === 'attributes') {
15                 window.trackedEvents[eventName].push(mutation);
16             }
17         }
18     };
19     window.observer = new window.MutationObserver(callback);
20     window.observer.observe(targetNode, config);
21 }, selector, eventName);
```

What kind of change events are we watching for?

This “mutation” is an animation event!

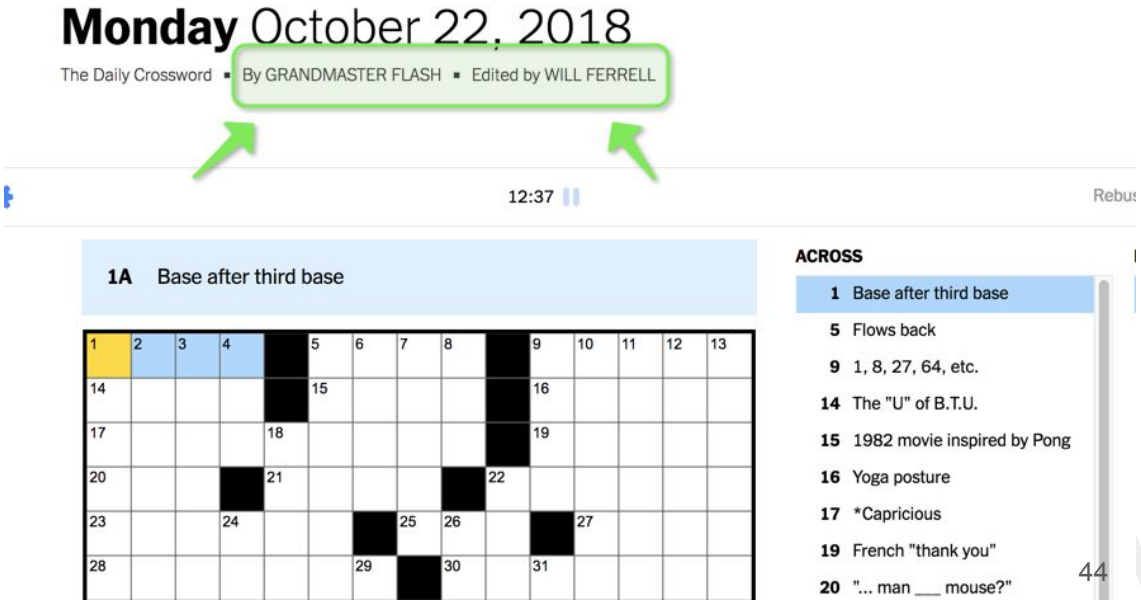
Set up a watcher for these animation events

```
1  const animationEvents = browser.execute( function () {
2      return window.trackedEvents;
3  })
4
5  assert(animationEvents[0] && animationEvents[0].className).equals('animation-thing')
```

Using Mock Data

Why Use Mock Data?

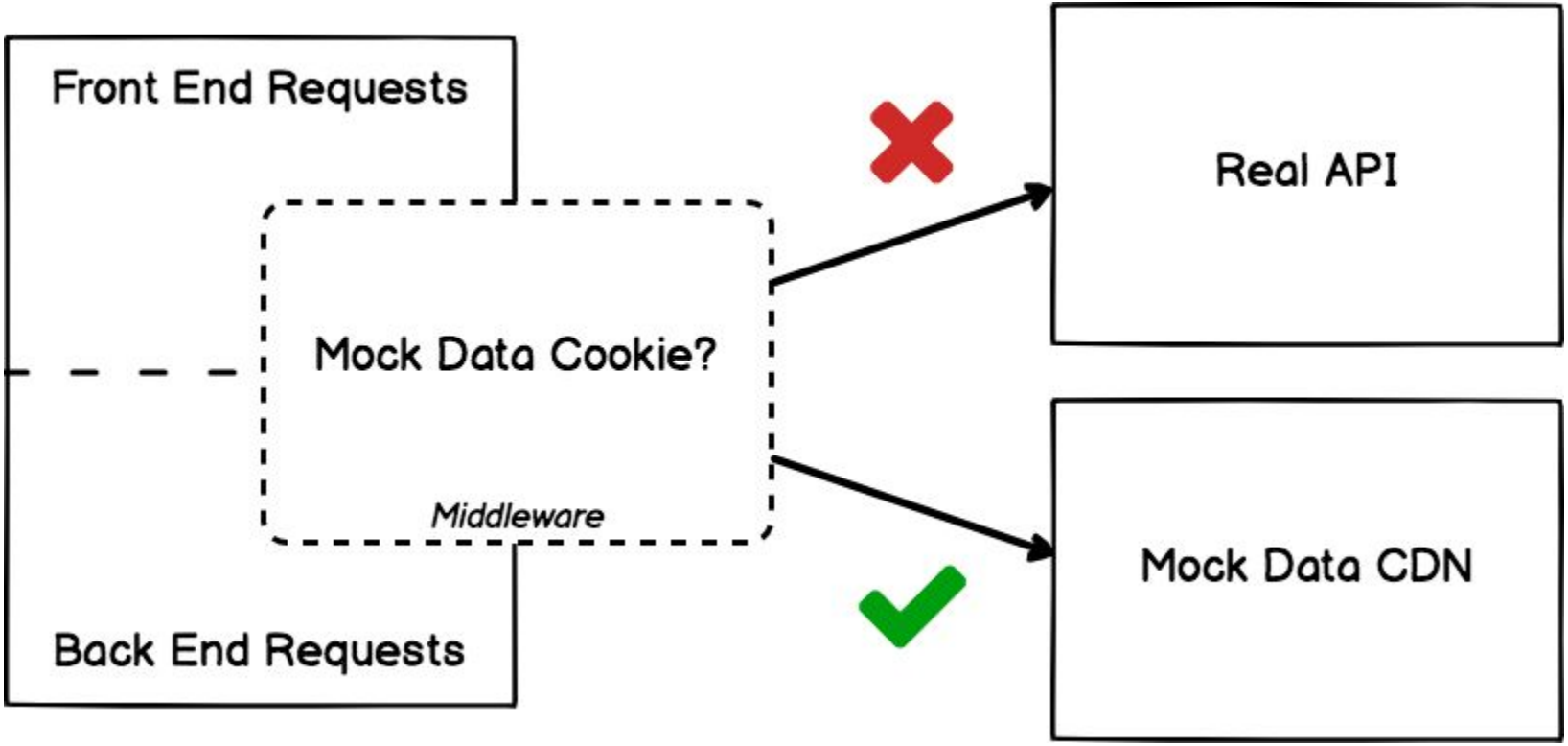
- API development not complete
- Hard to get the real API to deliver broken data for edge cases
- Set up visual diff states



We wrote a library!

- Express-sidetrack middleware
- Delivers mock data to our application
- Internal for now, OSS some day

How It Works


Front End Requests
Mock Data Cookie?
Middleware
Back End Requests
Real API
Mock Data CDN

Mock Data Gotchas

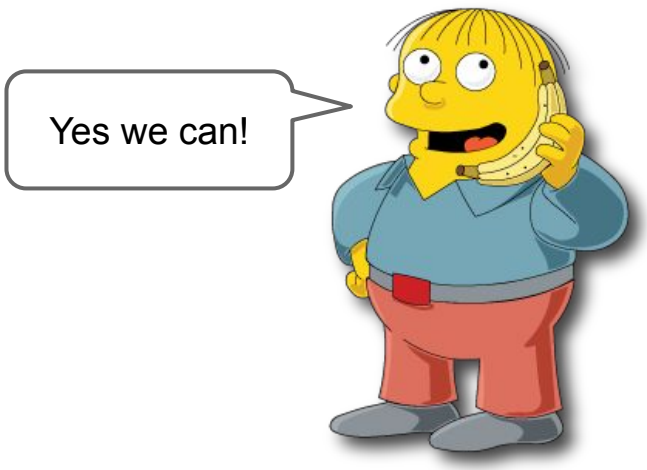
- Not a true end-to-end test
- Mock data have to be maintained
- Middleware has to be installed in the application under test
- Consider using an external proxy!

reliability

@thephilwells

Really Quite Fast Tests

Can’t We Fail Any Faster?



Really Quite Slow Tests

- “Regression testing the app across all the supported platforms will take 2 weeks.”
- “I made a little change but I can’t see if I broke anything without losing momentum.”
- “This single test takes a couple minutes to run on my local but times out after 5 minutes on Sauce Labs.”

SHAME

really quite fast

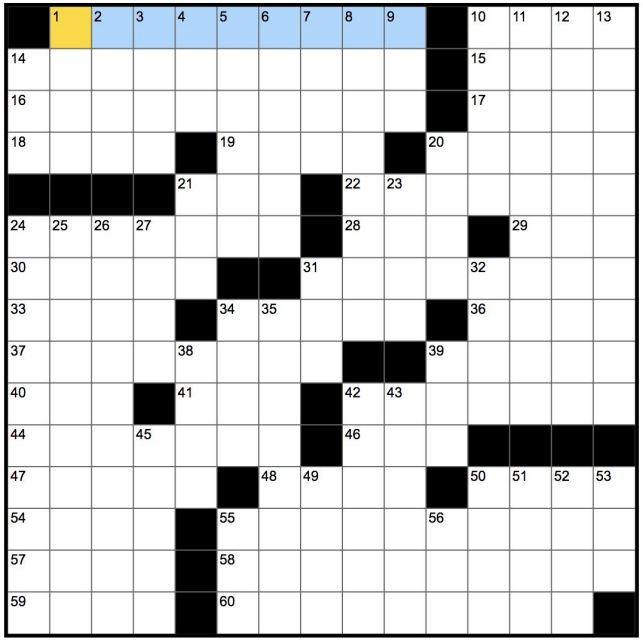
Disable Ads (most of the time)

- Slow to load
- Only explicitly tested in a few cases
- We can set a cookie to disable ads on test environments

really quite fast

Speeding Up Grid Interactions

Meet the Grid



- Made up of 144-225 cells
- Each cell contains 3 web elements
- Up to 675 individual web elements
- Some test cases require that all of the cells' states be validated

really quite fast

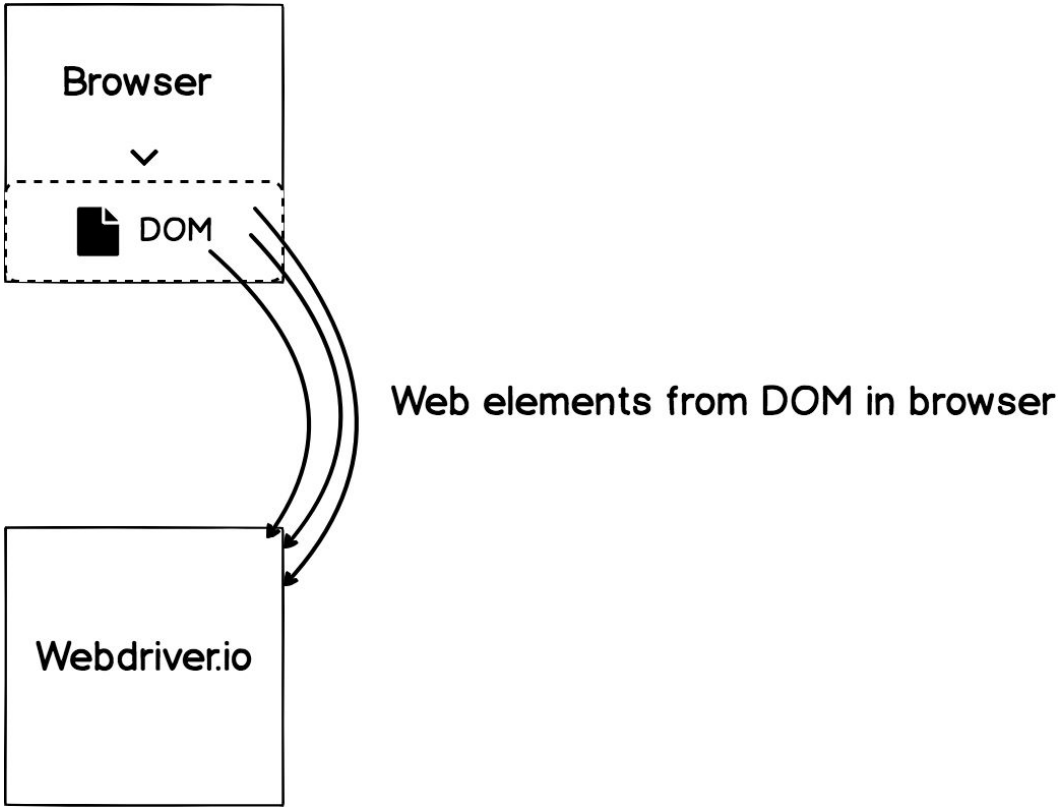
Traffic Jam

Operation	# of Network Calls	Seconds Network Traffic (m
Get all cells	1	0
For each c , get an ver ext	25	2 500
Now try it on Sauce VM's	255	***225,000!!!***

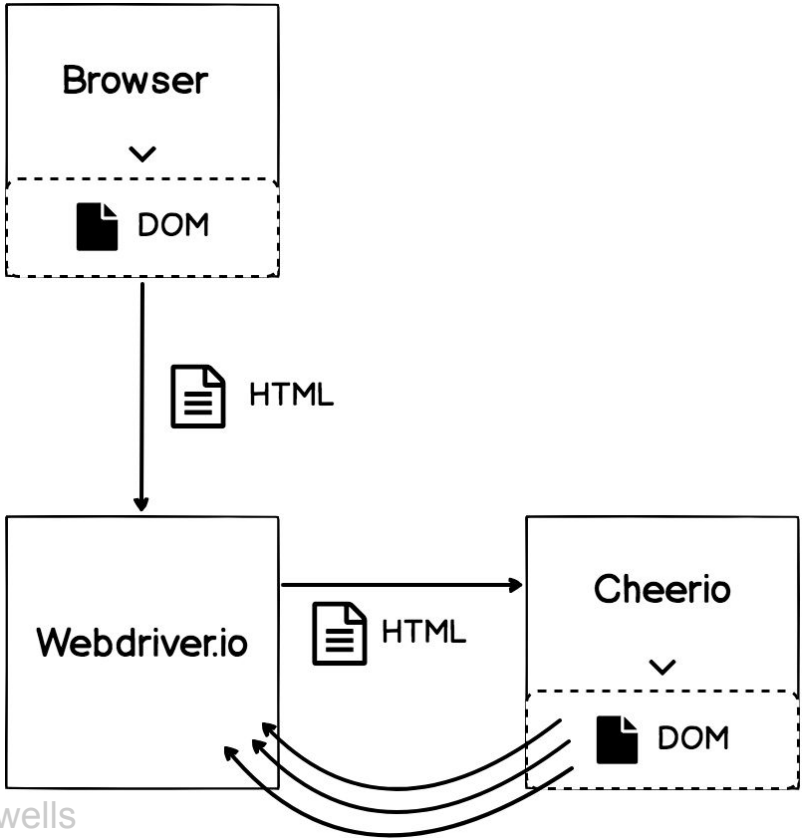
SHAME

really quite fast

Many Slow Requests



Do It Locally With Cheerio



Web elements from DOM in local memory

Cheerio.js Example

https://github.com/thephilwells/wdio-cheerio-demo

BUSTED!

How Are We Doing?

And What Work Remains

Measuring Our Progress

- Amount of rework
- Confident delivery
- Regression cycle length decreased, from days to hours

Support From Product Folks

- Making test automation part of our “Definition Of Done”
- Test effort considered during estimation
- The up-front cost of automation…

Embrace it!

Nobody's Perfect

- Team members have different comfort levels writing end-to-end tests
- All features ship with unit/component tests, end-to-end tests sometimes need to catch up post-release
- Need to capture ROI
- Tests still flake out sometimes!
- ***Software testing is a process of continuous shame elimination***

Questions?

There’s no shame in asking...

- https://www.linkedin.com/in/thephilwells/
- https://twitter.com/thephilwells
- https://github.com/thephilwells