



DrupalCamp Bristol 2015



SIFT
DIGITAL



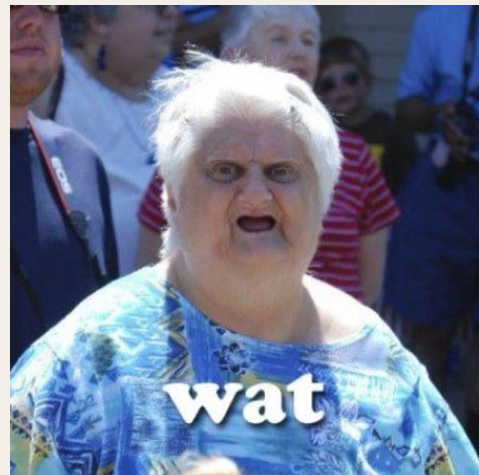
Automated front end regression testing and build pipes

Wat?

A poorly titled talk by Mike Dixon of ComputerMinds.

A walk through our “build pipe”.

Our approach to automated regression and visual testing.



Shameless plug - ComputerMinds



Build pipe - sounds cool - but why?

Builds are becoming increasingly complex.

Clean deployment becoming much more important.

Cannot rely on clients. (sorry clients)

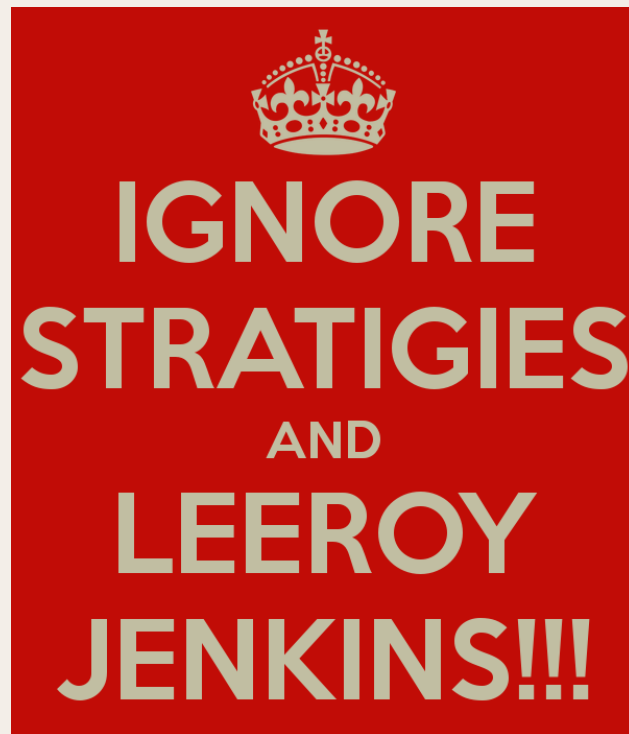
How is Sass compiled and managed.

Lets go - the build pipe.

Everything run through Jenkins.

Build job which executes on push to the project repo.

Job executes and the fully built site is then pushed into the "Full" repo.



The build job

It all starts with a make file

```
includes[stub] = "%PATH_TO_BUILD%/stub.make"  
projects[spire_profile][download][branch] = "%BRANCH%"  
projects[spire_profile][download][revision] = "%REVISION%"
```

The variables are replaced with parameters from the build job.



The stub make

```
core = 7.x
```

```
api = 2
```

```
projects[drupal][type] = core
```

```
projects[drupal][version] = 7.35
```

```
projects[drupal][patch][] = XXX
```

```
projects[yyy][type] = profile
```

```
projects[yyy][download][type] = git
```

```
projects[yyy][download][url] = git@github.com:computerminds/yyy.git
```

```
projects[yyy][download][branch] = ZZZ
```


The project repo



The project repo contains JUST the profile.

Includes custom modules and themes.

The project make file is used to fetch contrib modules and apply any patches.

The stub make file will fetch and run the project make.

Special sauce

A little bit of special sauce to:

Get the commit messages from the project repo and push into the full repo.

Allow files to be added at the Drupal root level.

Sassy grunting

Sass is compiled into CSS and committed into the full repo.

Grunt tasks run to do any Javascript monkeying.

Any other preprocessing?



Subliminal advert



The end result

We end up with a github repo containing a complete Drupal website, which can be deployed very simply with no complex dependencies.



Going forward

Composer FTW.



Regression testing

Many developers making many changes.

Much code and Sass re-use.

Many configurations and options.

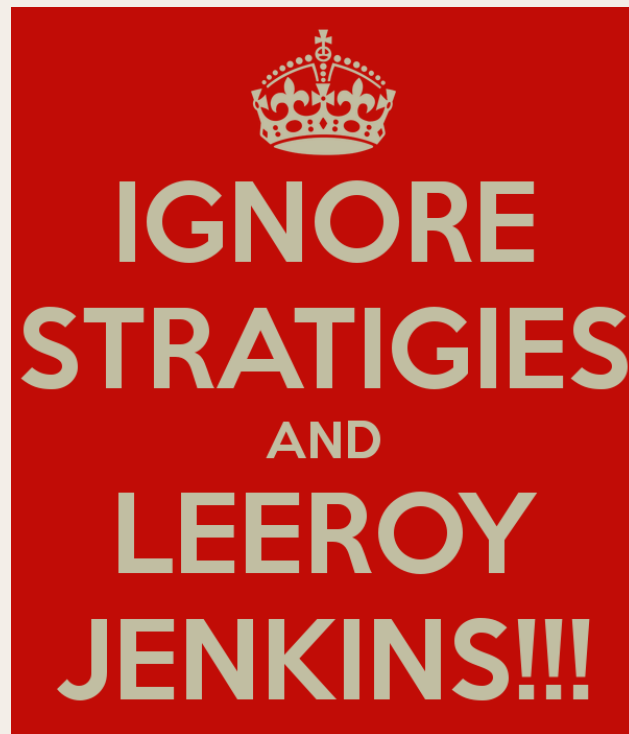
Not all options used all the time.

Time for more jenkins

Jenkins job triggered by the build pipe.

Run for specific branches.

Needs a deploy first, so we have something to test.



What are we testing

1. Code - unit tests - how useful are these for a typical web project?
2. Functionality - using headless browser to perform actions and check the response.
3. Visual - using a headless browser to perform actions and check the visual output.

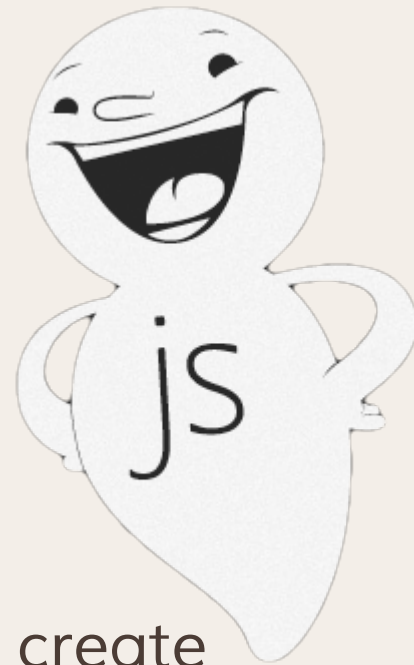
CasperJS

Scripting to control phantom and slimer.

Many buzzwords.

Create tests using coffee script. Option to create screenshots.

Target specific selectors.



An example test

```
casper.start "#{site}#{page}"  
selector = ".tile-type-media"  
  
test.assertExists selector, "The selector '#{selector}' exists"  
  
phantomcss.screenshot selector, "#{tile_name}-#{test_name}-D"
```

Very simple test - will just grab a screenshot of a specific selector.

Managing tests

The tests and the 'baseline' are stored in git.

Tests and baseline need maintaining.

Tests and baseline specific to a branch.

The process is only as good as the tests.

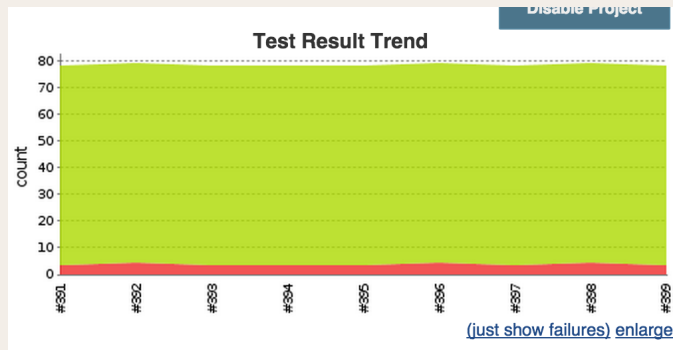
The output

Failed tests highlighted in Jenkins.

Email sent to the team.

Big red notification in hipchat.

Abuse and ridicule for the dev - sent to bed without dinner.

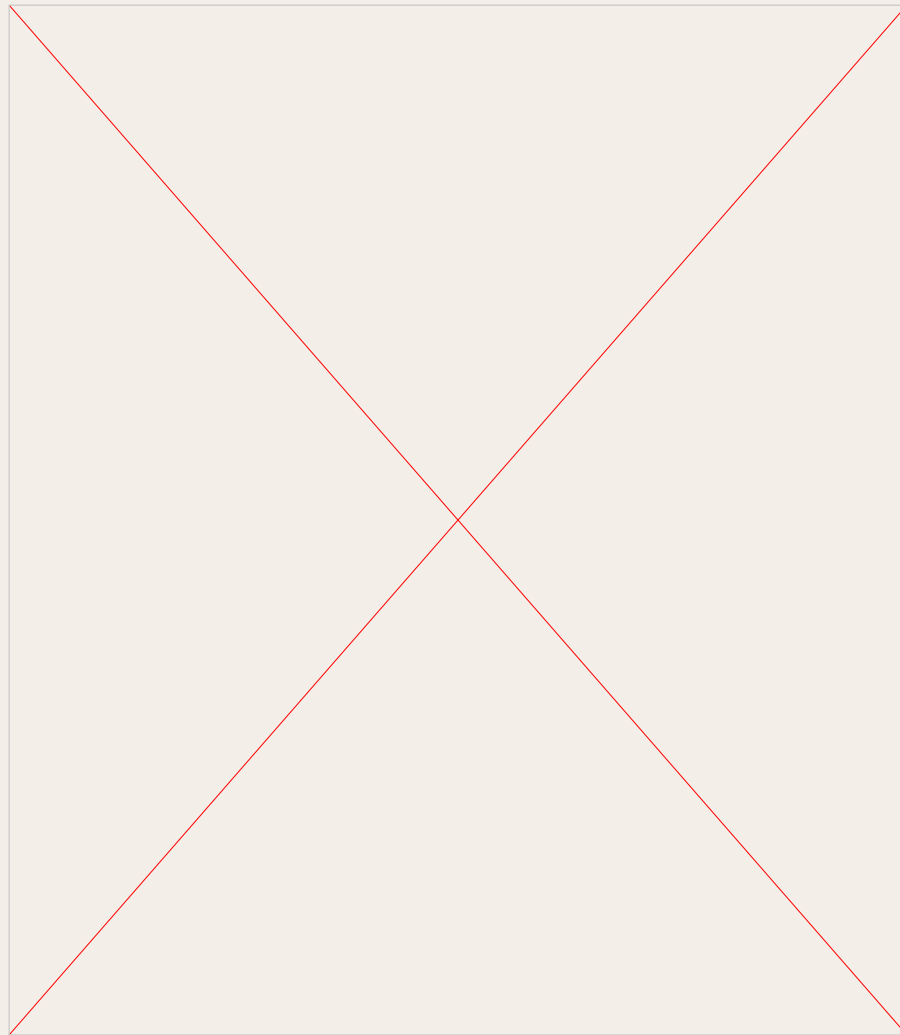


Visual comparison

It's relatively easy to automate the spotting of differences.

It's harder to see what those differences actually are, and what needs to be done to fix them.

We built Jeff.



Jeff

./screenshots/TABS-tile-accordion-M_18.png

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi in erat mollis, vehicula diam sit amet, fermentum leo. Praesent odio magna, tincidunt vitae libero vitae, ornare euismod nibh. Maecenas porta sagittis metus, et lacinia nulla euismod eu. Ut lorem dui, euismod eget velit et, fringilla pellentesque nisi. Praesent et dictum arcu. Aliquam condimentum libero vel purus consequat, quis sollicitudin dolor dignissim. Morbi lacinia cursus pulvinar. Phasellus egestas varius leo at scelerisque. Fusce accumsan turpis tempor eros bibendum ultrices. Mauris vel diam venenatis velit suscipit molestie pellentesque sit amet lectus. Cras suscipit malesuada risus, consectetur dignissim nisl imperdiet in. Sed vestibulum arcu elementum, sollicitudin erat ac, viverra urna.

[Click here >](#)

— Lorem ipsum 2 —

— Lorem ipsum 2 —

What else is out there?



Wraith - <https://github.com/BBC-News/wraith>

Ghost inspector - <https://ghostinspector.com/>

Applitoools - <https://applitoools.com/>

Huxley - <https://github.com/facebookarchive/huxley>

Any more? Let us know @computerminds

Next steps

Options to modify content - could be done simply using JS to squirt known text into markup.

UI for building the tests - then hand over to Danny D.

Make it easier to setup for new projects.

Create tickets in redmine.

Questions

Ask now?

Ask later :

Tweet @computerminds

mike.dixon@computerminds.co.uk

or find me on Strava :)



Slide deck available - will tweet URL with #dcbristol