Open Source Your JavaScript

Tooling and best practices

NickSullivan.prototype

VP of Technology at Krux - It's your data.

krux

Help organize SFJS* and 106 Miles in SF

creationeer.me

* Dave Nugent does all the work

Give me feedback in real time

Instead of "hmm"... +1 (by giving me a thumbs up) for stuff:

You learn

You find interesting

You want to know more about

Just like most sites nowadays...

I will be recording and collecting that data

You don't get to control how that data is being used...

AND

I will be selling it to advertisers later.

So before we go further...

Terms and conditions for this talk (standard stuff)

Do you see any Teletubbies in here? Do you see a slender plastic tag clipped to my shirt with my name printed on it? Do you see a little Asian child with a blank expression on his face sitting outside on a mechanical helicopter that shakes when you put quarters in it? No? Well, that's what you see at a toy store. And you must think you're in a toy store, because you're here shopping for an infant named Jeb.

Now that we know who you are, I know who I am. I'm not a mistake! It all makes sense! In a comic, you know how you can tell who the arch-villain's going to be? He's the exact opposite of the hero. And most times they're friends, like you and me! I should've known way back when... You know why, David? Because of the kids. They called me Mr Glass.

Do you see any Teletubbies in here? Do you see a slender plastic tag clipped to my shirt with my name printed on it? Do you see a little Asian child with a blank expression on his face sitting outside on a mechanical helicopter that shakes when you put quarters in it? No? Well, that's what you see at a toy store. And you must think you're in a toy store, because you're here shopping for an infant named Jeb.

My money's in that office, right? If she start giving me some bullshit about it ain't there, and we got to go someplace else and get it, I'm gonna shoot you in the head then and there. Then I'm gonna shoot that bitch in the kneecaps, find out where my goddamn money is. She gonna tell me too. Hey, look at me when I'm talking to you, motherfucker. You listen: we go in there, and that nigga Winston or anybody else is in there, you the first motherfucker to get shot. You understand?

However, the first round of drinks will be purchased by Nick Sullivan for everyone who accepts these terms and goes out after the talk

Normally, both your asses would be dead as fucking fried chicken, but you happen to pull this shit while I'm in a transitional period so I don't wanna kill you, I wanna help you. But I can't give you this case, it don't belong to me. Besides, I've already been through too much shit this morning over this case to hand it over to your dumb ass.

Do you see any Teletubbies in here? Do you see a slender plastic tag clipped to my shirt with my name printed on it? Do you see a little Asian child with a blank expression on his face sitting outside on a mechanical helicopter that shakes when you put quarters in it? No? Well, that's what you see at a toy store. And you must think you're in a toy store, because you're here shopping for an infant named Jeb.

Now that we know who you are, I know who I am. I'm not a mistake! It all makes sense! In a comic, you know how you can tell who the arch-villain's going to be? He's the exact opposite of the hero. And most times they're friends, like you and me! I should've known way back when... You know why, David? Because of the kids. They called me Mr Glass.

Do you see any Teletubbies in here? Do you see a slender plastic tag clipped to my shirt with my name printed on it? Do you see a little Asian child with a blank expression on his face sitting outside on a mechanical helicopter that shakes when you put quarters in it? No? Well, that's what you see at a toy store. And you must think you're in a toy store, because you're here shopping for an infant named Jeb.

My money's in that office, right? If she start giving me some bullshit about it ain't there, and we got to go someplace else and get it, I'm gonna shoot you in the head then and there. Then I'm gonna shoot that bitch in the kneecaps, find out where my goddamn money is. She gonna tell me too. Hey, look at me when I'm talking to you, motherfucker. You listen: we go in there, and that nigga Winston or anybody else is in there, you the first motherfucker to get shot. You understand?

[ ] Please raise your hand and "tick" to accept

Just kidding

For those who accepted:

```
"The first round of drinks will be purchased by Nick
Sullivan for everyone who accepts these terms and goes out
after the talk."
```

First drink on me!

Why am I doing this?

I'm passionate about open source, engineering productivity, and tooling.

Creating my own open source projects, postscribe, pilotfish.io - I went looking for a best practices list.

Scanned other open source projects to see what they were doing. jQuery, backbone, bootstrap, and others.

Project idea at the end.

And maybe I'm a little OCD

I'm starting group meetings at my house for people with OCD. Not because I have it, but surely one of them will be bothered enough to clean it.

Why Open Source Your JavaScript?

The good reasons

Help other people

Turns you into a producer, not a consumer

Improves your project

- More usage
- More feedback
- More contributors

Guiding principles

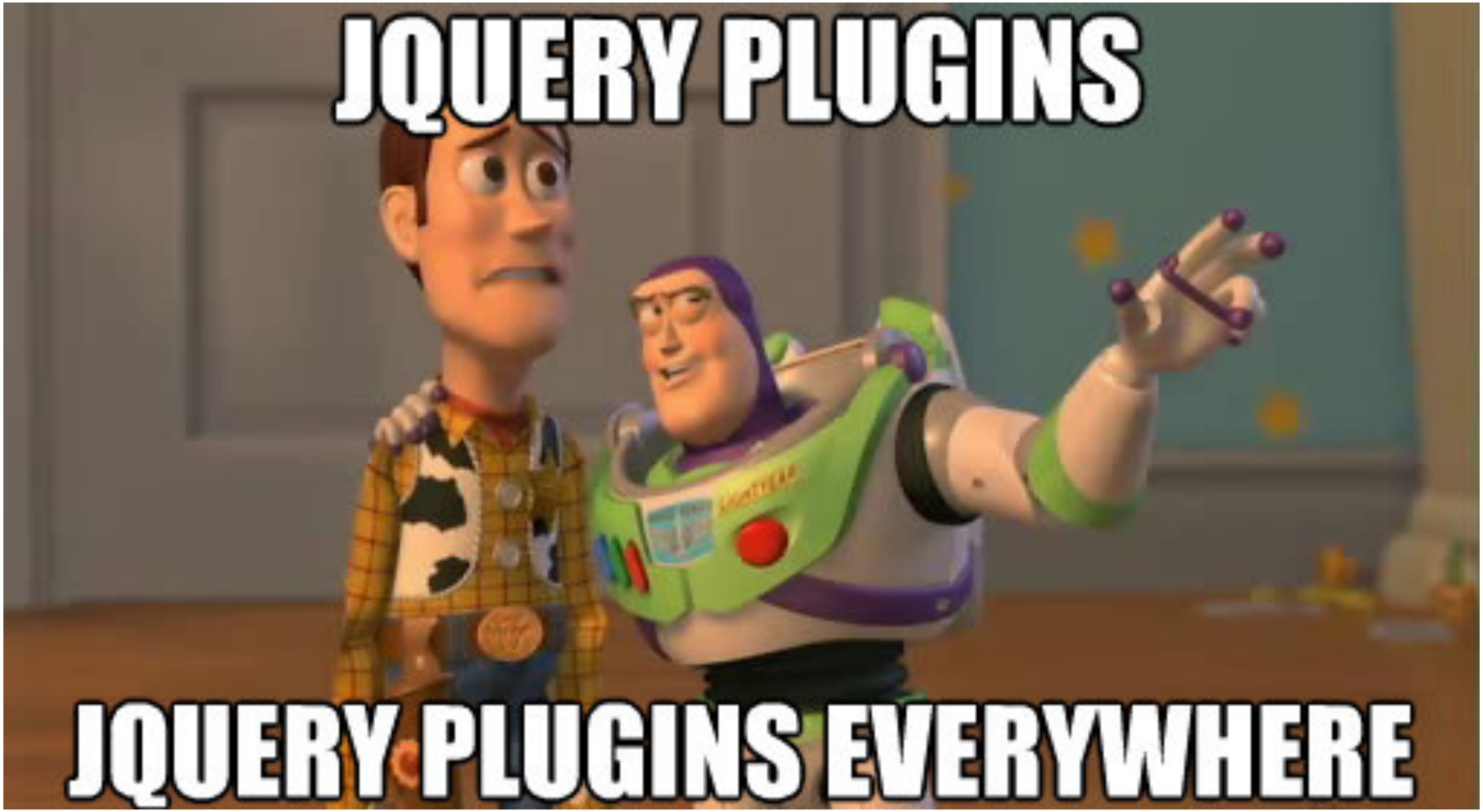
Increase quality

With: *Repeatable processes* to increase software quality. Tooling for code creation, testing, and release.

Encourage usage

With: Great documentation, demonstrations, and communication

Current Landscape



Why is there fragmentation in JS?

"It doesn't do *exactly* what I want"

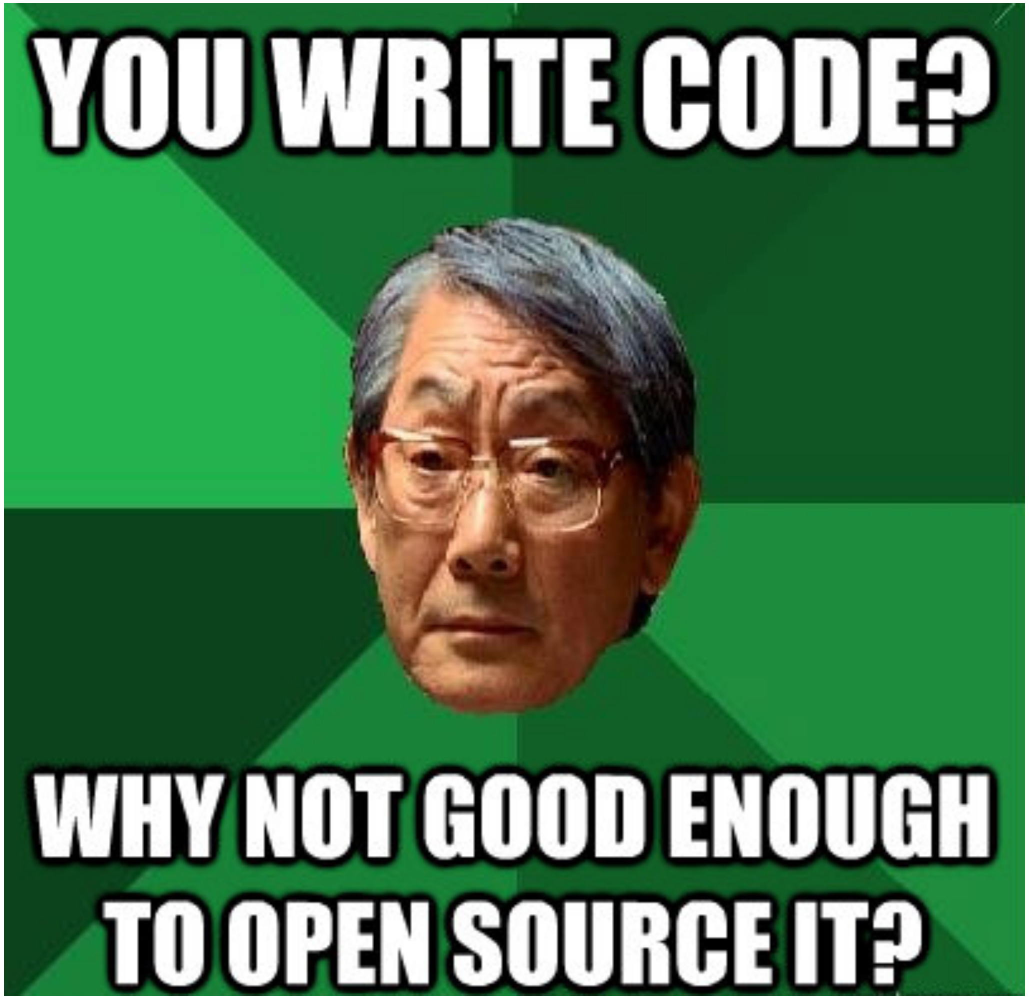
"I couldn't get it working" *aka* "The documentation sucks"

Net - **it was easier to just rewrite myself.**

Javascript is easy - lowers the bar for rewriting.

Which means we need to raise the bar for ease of use and implementation. **Let's fix this.**

On Quality



Quality - Internal vs Open Source

Good enough isn't... good enough.

It's the ultimate code review. Your reputation is on the line.

Most of these concepts apply to internal software too, we just don't take the time. Let's make that easier.

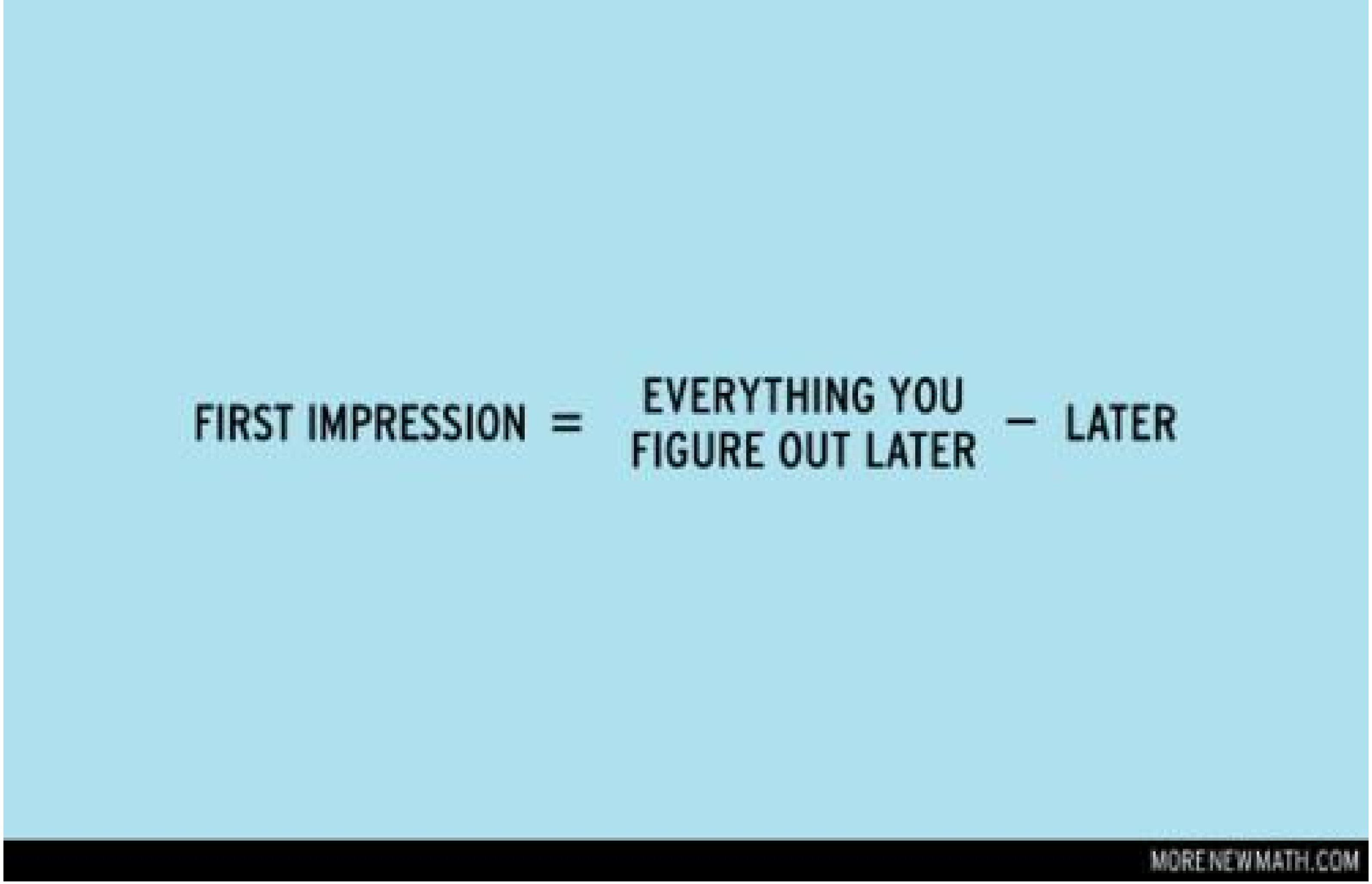
But we balance quality with speed

A quote on *when* to release.

When you start community-building, what you need to be able to present is a *plausible promise*. Your program doesn't have to work particularly well. It can be crude, buggy, incomplete, and poorly documented. What it must not fail to do is (a) run, and (b) convince potential co-developers that it can be evolved into something really neat in the foreseeable future.

- Eric S. Raymond

First Impressions



FIRST IMPRESSION = EVERYTHING YOU FIGURE OUT LATER − LATER

MORENEWMATH.COM

Make it count

Developers review dozens, if not hundreds of projects per year. From this experience we filter out projects that:

- Don't work
- Have poor documentation
- Can't get support for
- Seem immature/unpolished
- Have low activity

Your project won't likely get a second chance.

More from thechangelog.org - http://tinyurl.com/4ec4lpx

Two different first impressions

End Users - README.md, /docs, or minisite

Contributors - DEVELOPMENT.md, CONTRIBUTING.md, and TODOS.md

What end users are looking for

Your README is your marketing doc

Potential end users are looking for:

1. Will it do what I want?
    a. Good description of the *problem* and *solution*
2. Does it work?
    a. Can I see a demo? Is the repo active?
    b. Can I see it live on a real site (who is using it?)
3. Will I be able to get support?
    a. File a bug report? Are they getting fixed?
    b. Mailing list? Is the community active?

What contributors are looking for

1. How do I get the code running?
   a. Development environment. **< 15 minutes**
   b. Running tests
2. What is the process for submitting changes?
   a. Pull requests
   b. Branching strategy
   c. Style guides
   d. Rules
3. What is the release process?
4. What else can I help with?
   a. Public TODOs
   b. Github issues

Repo Setup

2 hard things in computer science

1. Cache Invalidation

2. Naming things

I propose a 3rd -

3. Asynchronous Programming

#1 and #2 Phil Karton. #3 Nick Sullivan.

Choosing a name

SEO

What will *users* be searching for solve their problem?

Good: "node-geoip-server". *Bad*: "$futurama_character"

Is there an existing github project/organization by that name? github/$name

Other things to check for availability

- domain name, twitter, facebook

Tip: ${projectname}js is a growing trend

Github organizations

Project big enough for more than one repo?

http://github.com/$organization/$project1
http://github.com/$organization/$project2

https://github.com/blog/674-introducing-organizations

README driven development

Organize your thoughts around what you want to build, in present tense.

The documentation evolves with the product - instead of being an afterthought.

Cater to your initial audience - write the readme as you would want to see it if you were a user.

http://tom.preston-werner.com/2010/08/23/readme-driven-development.html

README format

Use markdown, not plain text.

Tip - Mou for mac

Note that github uses github flavored markdown

Anyone from github here?

Anyone want to build a web-based markdown editor that displays github markdown in real time?

What should be in your README

- About*
- Getting started / Download / Usage*
- Dependencies
- Who is using it? (USING.md)
- Where to go for help
- How to contribute
- Licensing

* = Cater to your initial audience. Get up and running as quickly as possible.

Repo files

NPM

Make your javascript project a node package - even if it is browser only. Lets you use npm, travis, and other tools for automation.

How?

Set up a package.json. That's it!

Optional:

```
$ npm publish
```

Tip: `private: true` in your package.json to prevent it from being published to npm registry

Package.json validation

Surprisingly, no easy way to do this.

Multiple standards

- CommonJS Packages 1.0
- CommonJS Packages 1.1
- Nodejitsu
- NPM (use this one)

http://package-json-validator.com/
(I created last weekend)

Repo directories / code organization

Standards for directories (from Packages 1.0 spec)

- /src
- /doc
- /lib - external libraries
- /build - doesn't really apply to js
- /dist - released versions
- /test - unit and functional

/docs and /tests ok, but be consistent

Licensing

LICENSE or LICENSE.txt

Masochists:

http://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses

MIT the most open; BSD requires credit.

New: WTFPL license (do whatever the f*ck you want)

Consistent formatting

`.editorconfig` - definitions of spacing for the project. Stop fixing other peoples formatting

Consistency is more important than preference

Allows for definitions by file for .js, .html, etc.

Supported by the editor - I couldn't get it to work with vim. :(
More at editorconfig.org

Gitignore

.gitignore

```
node_modules # never check these in
.*.swp #vim
npm-debug.log
```

Exhaustive - html5boilerplates .gitignore

Release notes

Release notes build trust. Recommend format:

```
### Version 0.2 - Date
* [BUG] Fix
* [NEW] Feature
* [API CHANGE]

### Version 0.1 - Date
...
```

Either in README or separate CHANGELOG*, CHANGES, or HISTORY file

Contributor Workflows

Productivity for everyone

Repeatable process

Increase your productivity and avoid missed steps by letting the computers do repetitive tasks (they are good at it)

grunt.js is awesome

- Minification
- Release
- Linting
- Testing
- More!

Sample gruntfile for pilotfish

Version numbers build trust

Semantic versioning - semver.org, node-semver, APR version guide

Major.Minor.Patch.Trivial+build

- Major - Backward incompatible changes
- Minor - New Functionality
- Patch - Fixes
- Trivial - Documentation, formatting
- Build - build id

Put version in the code! As part of your build!

Branching/Tagging strategy

Common approaches:

1. Work in master. Branches for features.
2. Work in dev branch. Merge to master for release.

#1 works best, particularly when you consider forking and pull requests. For familiarity and ease of use within the community use #1 unless you have a good reason not to.

Tip: Tag release/$version (with repeatable process so it's consistent - use grunt!). Github tag = downloadable .zip - sample

Static analysis

jshint / jslint

I prefer jshint, better integration with editor
.jshintrc - sample from pilotfish

When to run:

- on every save with your editor
- grunt lint
- part of integration tests

JSHint

Production/Debug versions

Minification with Uglify - via grunt

Consider permanent urls with version numbers (also done with grunt)

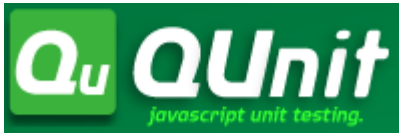
- /1.8.3/pilotfish.min.js
- /1.8/pilotfish.min.js
- /1/pilotfish.min.js
- /stable/pilotfish.min.js
- /latest/pilotfish.js (debug)

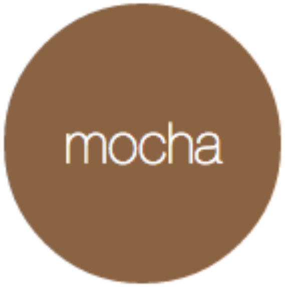
Testing

If it's not tested, it's not done

Unit tests

Run in the browser, manually. Main two I found being used:





QUnit FTW.

Sample

Phantomjs

Run tests on *command line* with a webkit-based browser.

Awesome. No more Firefox in xvfb!

phantomjs.org

Casperjs

Thin layer on top of phantomjs

Completely emulate browser behavior. Clicks, submit forms, back, reload, refresh.

Run on command line with grunt-casperjs

Run tests across multiple browsers

browserling / testling

browserstack

testem

Detour - let's kill Selenium

Casperjs much easier to work with and much more robust, but it is not cross browser.

Can we get browserstack / browserling to read casperjs files and have them run against all browsers?!

Travis-ci!

Distributed continuous integration testing - Hosted Jenkins - without the suck.

Github webhook. Every commit, run tests. Sample travis.yml for pilotfish

Phantomjs is installed, casperjs isn't - see pilotfish travis.yml for pre install script

travis-ci.org

Github/Travis integration

Test all branches - you can see if a branch passes the tests before merging, *right in the pull request.*

Testing recap

- Unit tests
- Integration tests
- Across multiple browsers with browser stack
- Kill selenium
- Continuous Integration with Travis

Generated documentation

What's the only thing worse than no documentation?

Documentation that is wrong!

docco - markdown is in the js
Sample from pilotfish

Fire it off as part of your build/release process in grunt.

Github Pages for a mini site

If you are using a github organization, you can create a repo called $organization.github.com, and this will be served from http://$organization.github.com

- Faster than raw.github.com, because it is generated flat files instead of pulling from the repo
- Great for demos

Also see husky

Community

Simple philosophy
Don't be a dick

I'm not a community expert, and that's a talk in itself, but at least consider these for communication:

- Mailing List
- IRC

Repo Health

Idea that I'm working on an automated tool to scan this stuff for you. Starting point: Pilotfish rating scale

Service would scan, and make recommendations. Like Yslow for repos.

Some elements common for all projects, but plugin driven support for other languages needed.

I have the readme written :)
https://github.com/gorillamania/repo-health
See me after if you want to help

;

Go out and open source your code!

First drink on me.

Find me and these slides at creationeer.me