#### Let Humans Human and Computers Computer

A Tale of Building Useful and Effective Automations

Jan Krutisch <jan@depfu.io>

rubyconference.by 2017

#### Let Humans Human and Computers Computer

A Tale of Building Useful and Effective Automations

Jan Krutisch <jan@depfu.io>

rubyconference.by 2017

#### Building automations

#### Why automate?



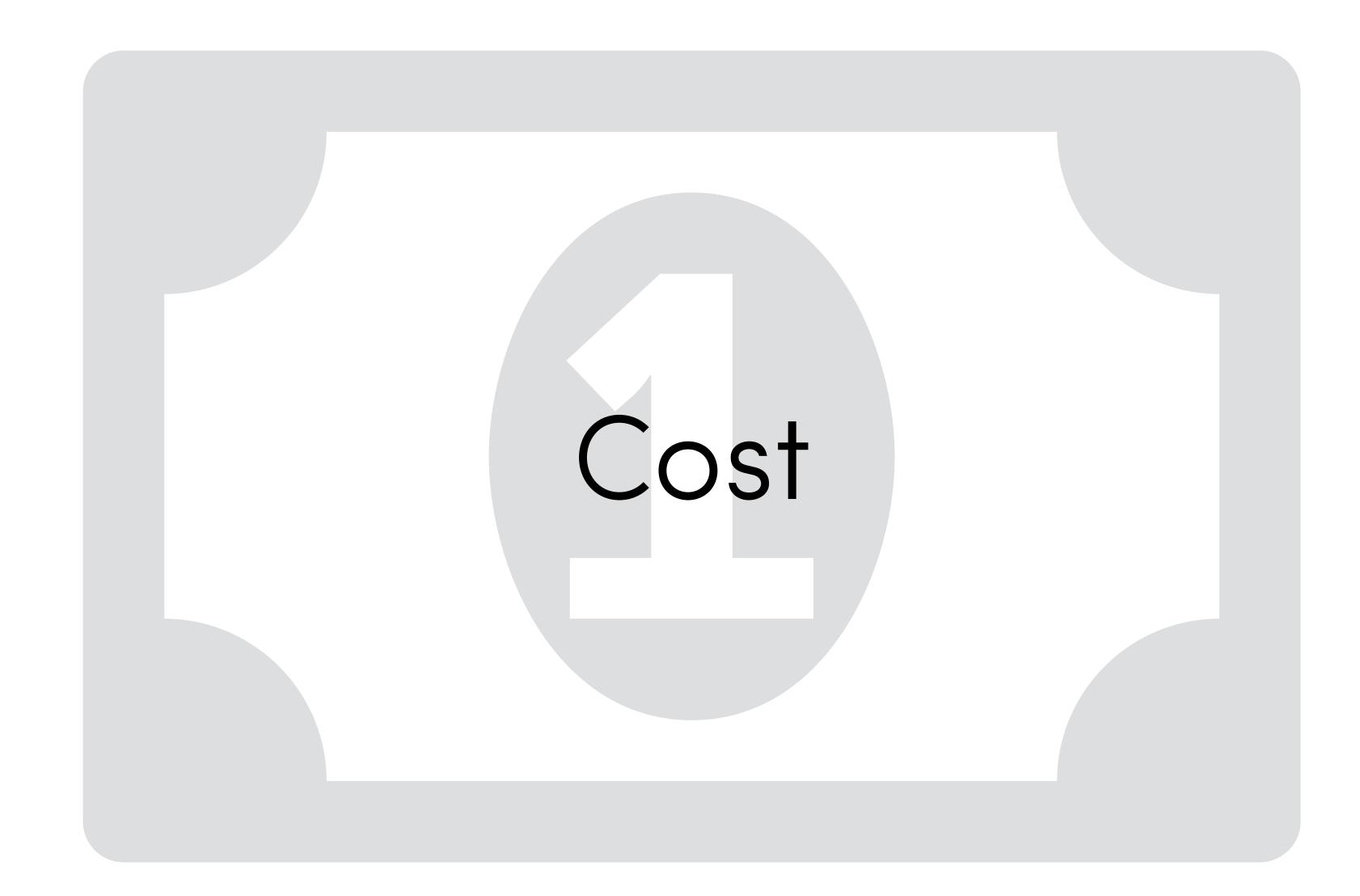
















# work more enjoyable less errors less cost and/or work

#### Downsides

# Testing

# Deployment

#### Dependency Updates

#### Testing

Ĺ

Ĺ

# Boring

# Repetitive

### Error prone

#### Unit tests

# Speed

#### Code coverage

#### Lesson #1

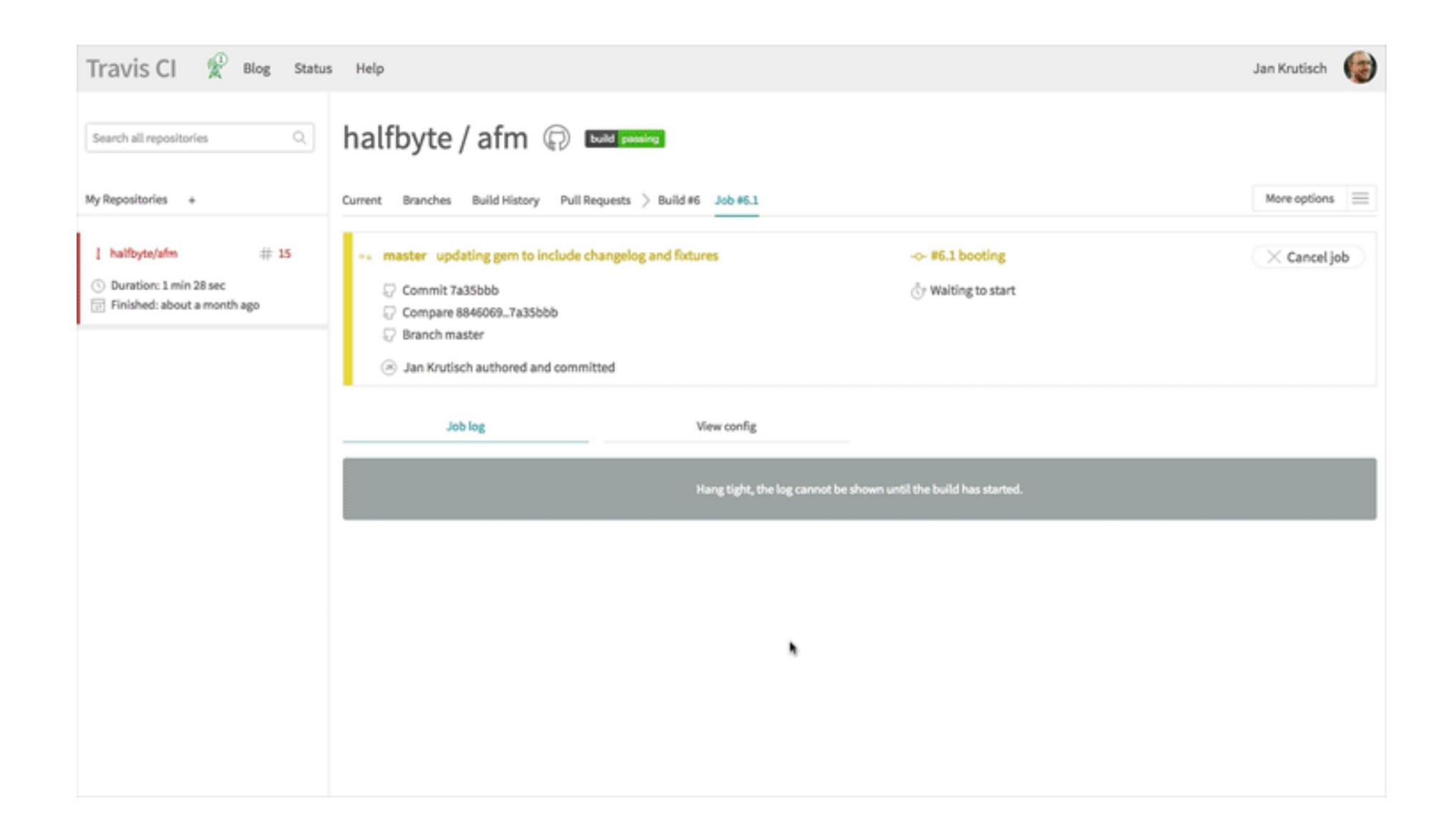
Good automations make the easy trivial and the complicated possible

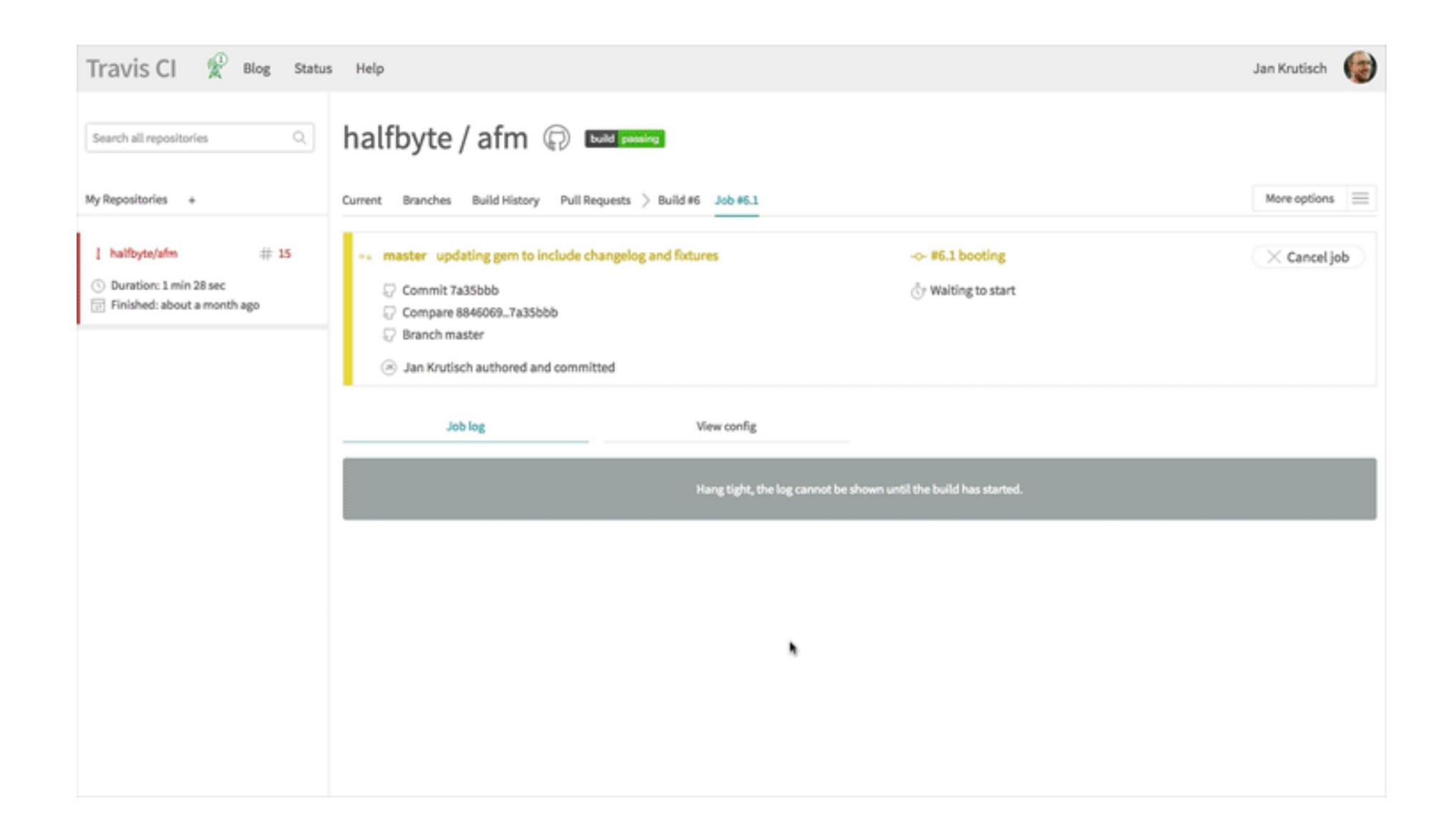
## Triggering

#### Lesson #2

Not everything that can be automated should be automated.

#### Continuous Integration



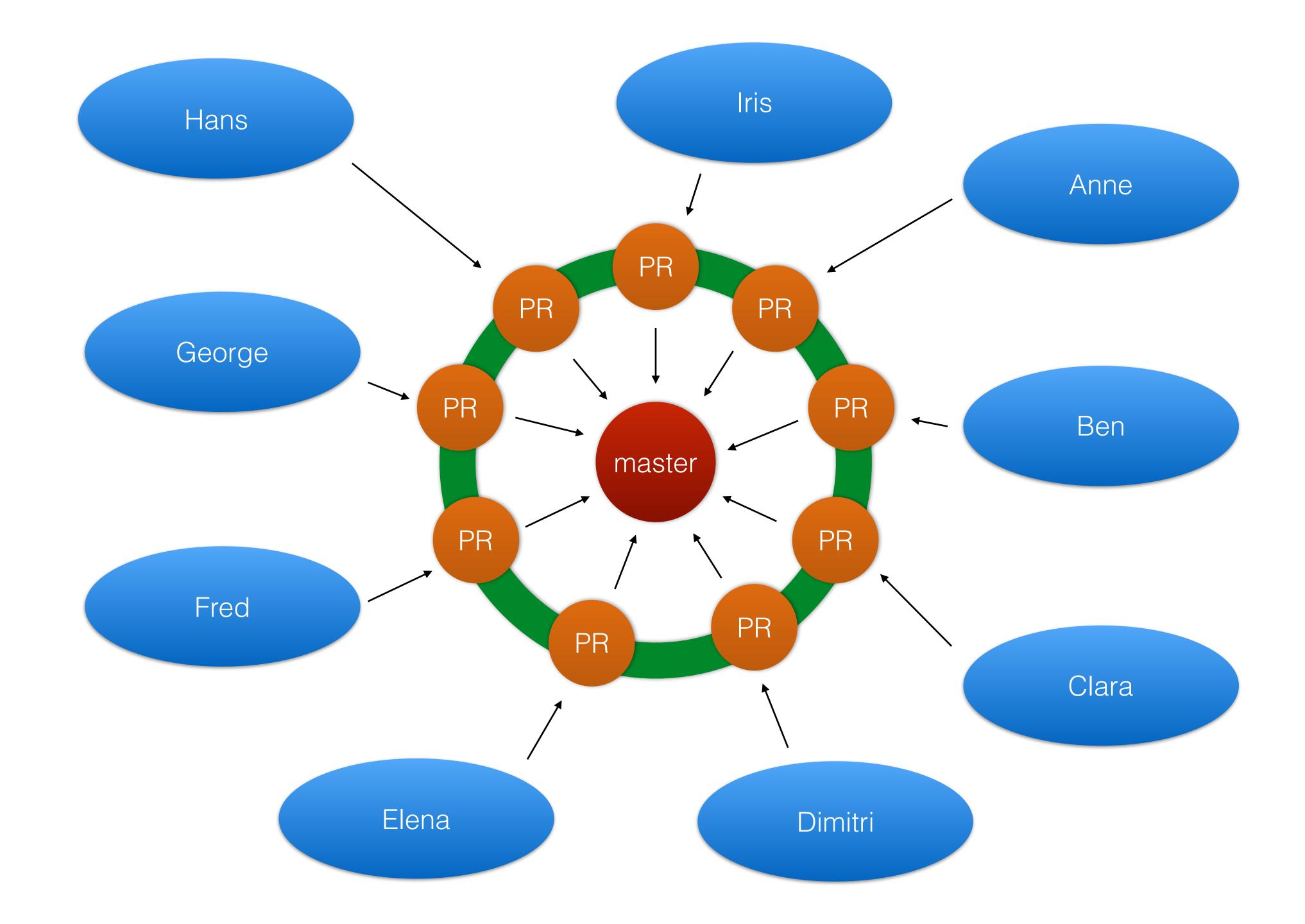






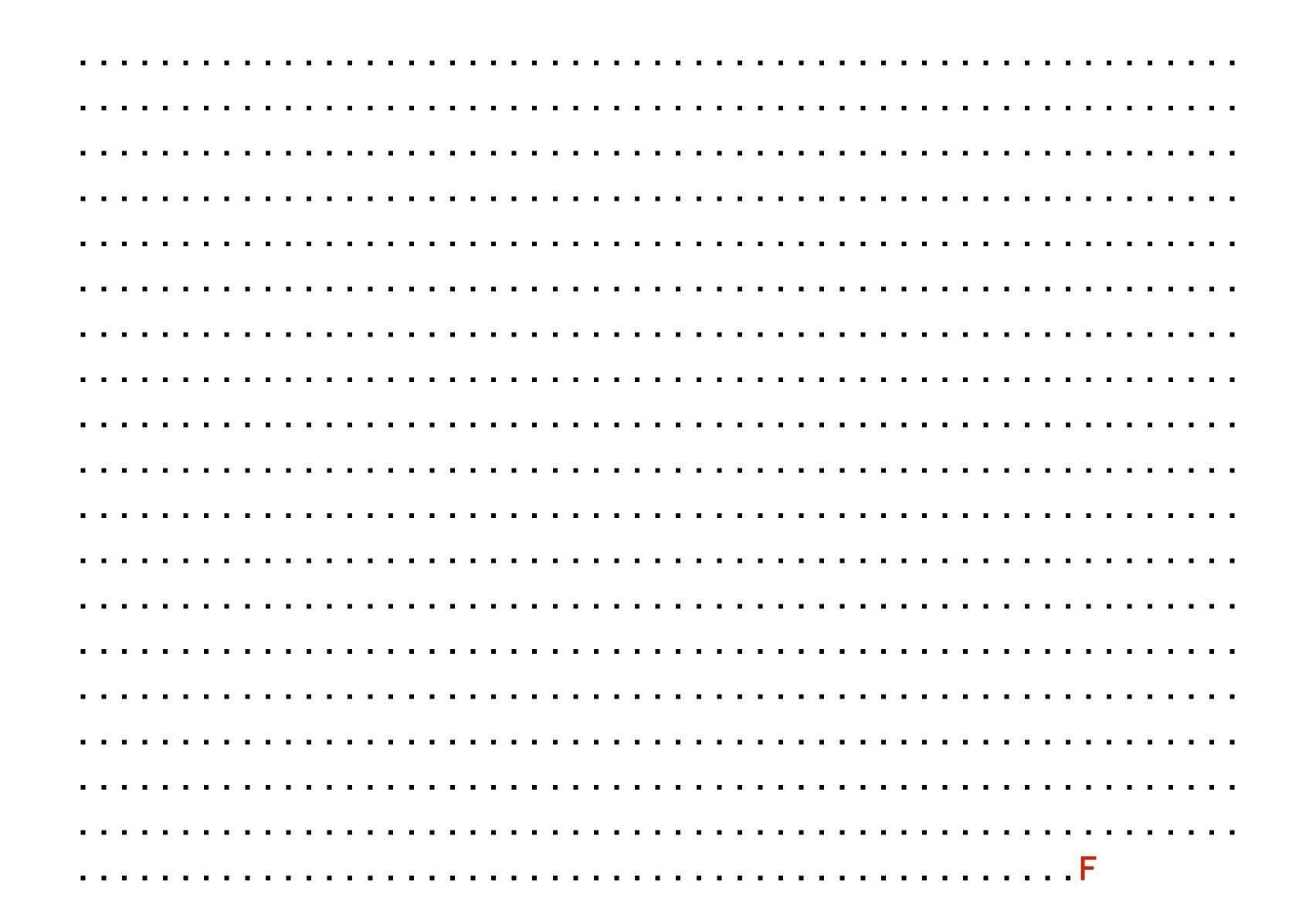
#### Human error

#### Continuous Integration



#### Lesson #3

Good automations integrate into your team workflows



#### High level tests

#### Cucumber

### Ojavascript

### Cl is asynchronous

#### Lesson #4

Asynchronous automations are a great way to unblock your developer workflows.

#### Deployment

jan@adora ~/Documents/projects/artdoxa <single\_artwork\_page>
\$ | | |

jan@adora ~/Documents/projects/artdoxa <single\_artwork\_page>
\$ | | |

#### FTP

mysql> UPDATE transactions ADD
COLUMN original\_value INT...

# Capistrano

# Boring

# Repetitive

# Error prone

# Speed

# Atomic operation



#### Lesson #5

Good automations codify best practices

#### Continuous delivery

## Continuous deployment

# Higher risk

# More complexity

#### Lesson #6

The higher the risk for a certain task is, the more complex your automation will have to be

#### Dependency Updates

\$ bundle outdated

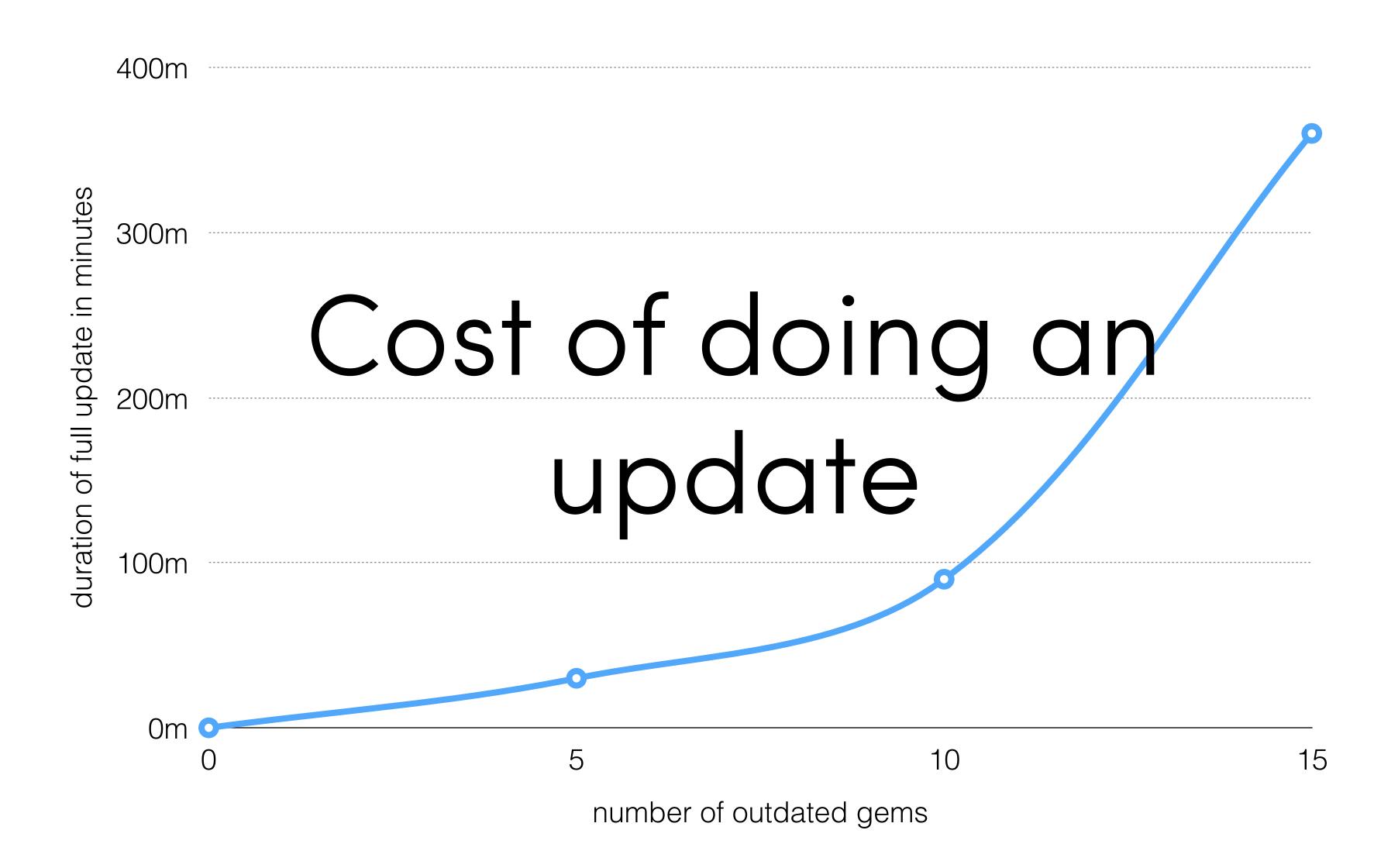
#### gemsurance

\$ bundle audit

#### Problem?

# Stability

# Security



### Pain in small doses

## Best practice?

### One at a time

## Find information

## green tests

## rinse and repeat

### make it better!





### One at a time

## New branch

## Tested by CI

## Find information

#### ▼ Commits

The new version differs by 27 commits:

- Prepare for 1.14.0 release
- Merge pull request #868 from teampoltergeist/extension\_failure
- Merge pull request #869 from teampoltergeist/header\_logic\_fix
- fix changelog [ci skip]
- Raise error when extension can't be loaded Fixes issue #867
- fix permanent header logic and just pass options through
- Merge pull request #857 from Nix-wie-weg/remove\_temp\_header\_on\_redirect
- Added 'permanent: :no\_redirect' to also remove temp headers on redirects
- Update CHANGELOG.md [ci-skip]
- simplify blocked requests and store in networkTraffic
- Merge pull request #859 from kylerippey/add\_blocked\_network\_traffic
- Began tracking blocked network requests and exposed it through the network\_traffic interface
- Merge pull request #856 from teampoltergeist/equal
- Node equality needs to also compare page\_id
- fix potential error
- fix test for JRuby 9.1.x.x
- test with jruby 9.1.7.0
- fix duplicate typo
- fix typo
- limit ttfunk version when testing with ruby versions < 2.1
- Merge pull request #853 from calasyr/patch-2
- Merge pull request #852 from calasyr/patch-1
- Added CircleCI to list of CIs that have PhantomJS installed by default.
- Updated README with link to latest version.
- Merge pull request #850 from teampoltergeist/evaluate\_element\_result
- Support returning structures containing elements from evaluate\_script
- Support returning an element from evaluate\_script

See the full diff.

#### 3.1.0 (27 February 2017)

- update UglifyJS to 2.8.0
- add reduce\_vars compress option
- enable reduce\_vars and collapse\_vars compress options by default
- unused top-level function and variable removal with toplevel compress option
- add top\_retain compress option to specify list of top-level variables to always retain
- add unsafe comps and unsafe proto options

#### 3.0.4 (30 November 2016)

- update UglifyJS to 2.7.5
- implement ignore\_quoted and debug options for mangle\_properties

#### 3.0.3 (24 October 2016)

- update UglifyJS to 2.7.4
- add wrap\_iife output option to wrap IIFEs in parenthesis

#### 3.0.2 (20 August 2016)

- add top-level keep\_fnames option that implies both compressor and mangler keep\_fnames
- update UglifyJS to 2.7.3

#### 3.0.1 (28 July 2016)

- update UglifyJS to 2.7.0
- split JS dependencies to separate files

#### 3.0.0 (22 March 2016)

- drop support for Ruby 1.8
- remove json as dependency
- discard unused function arguments only in unsafe mode
- add keep\_fnames option to preserve function names in compressed code
- add collapse\_vars option to collapse single-use variables
- backwards incompatible changes to source map options
- support for inline base64 encoded source maps
- mangle property names option (disabled by default)

○ v3.1.5 -0-18626f5

#### will\_paginate 3.1.5

mislav released this on 15 Oct 2016

- Avoid Rails 5 deprecation warning about original\_exception
- No longer generates <a rel="start"> for first page link since "start" is an invalid rel value

#### Downloads

- Source code (zip)
- Source code (tar.gz)

on 15 Oct 2016

v3.0.11 ... → 231a09e 🖺 zip 🖺 tar.gz

#### will\_paginate 2.3.17

- Mislav released this on 15 Oct 2016 · 359 commits to master since this release
- Avoid unintentional SQL queries being triggered by defining respond\_to\_missing? alongside method\_missing?

#### **Downloads**

- Source code (zip)
- Source code (tar.gz)

♦ v3.0.10 •• c993739

#### will\_paginate 3.0.10

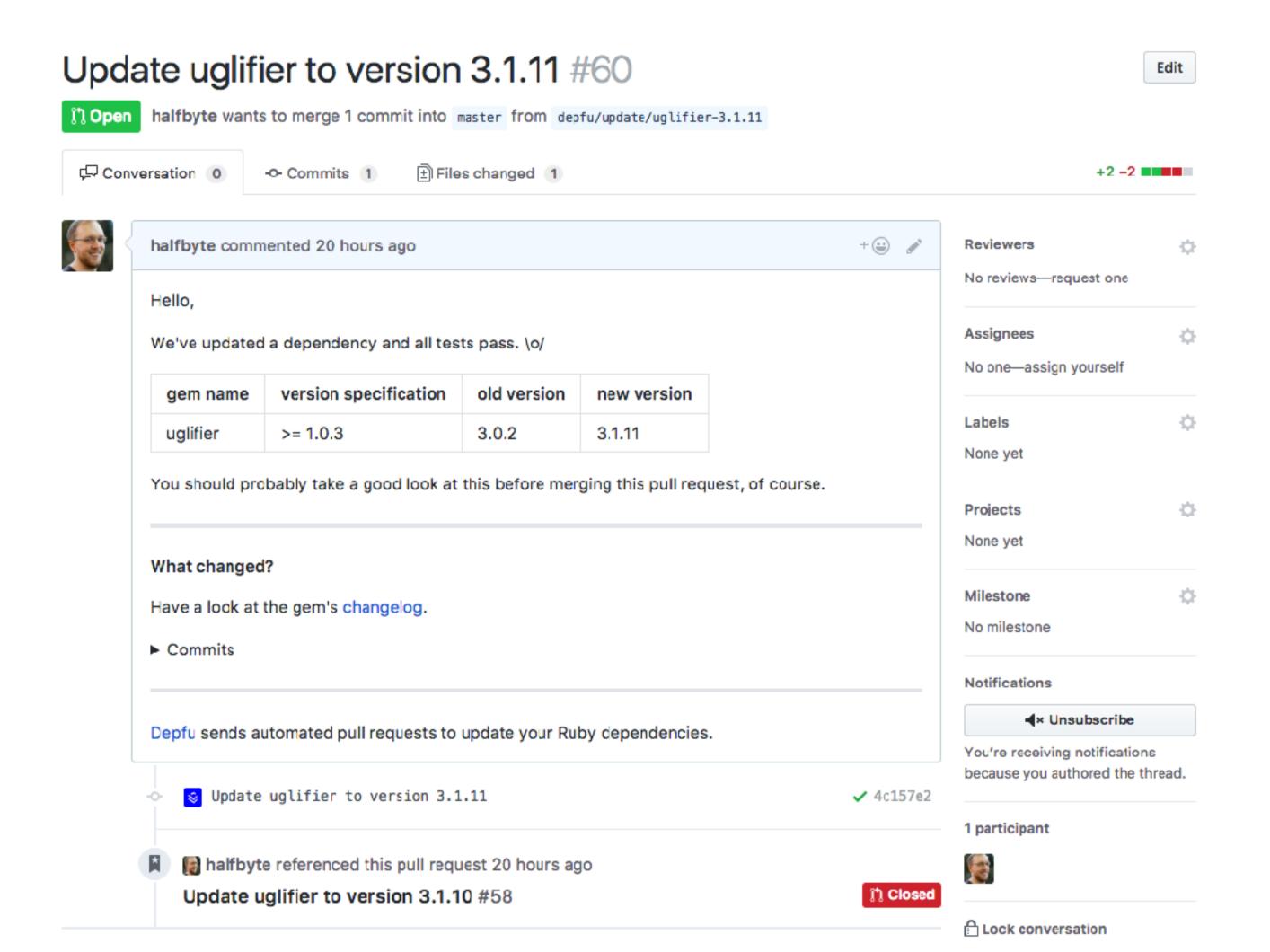
- mislav released this on 21 Sep 2016 · 108 commits to master since this release
- Fix locale load path so that application can override will\_paginate's built-in translations.

#### **Downloads**

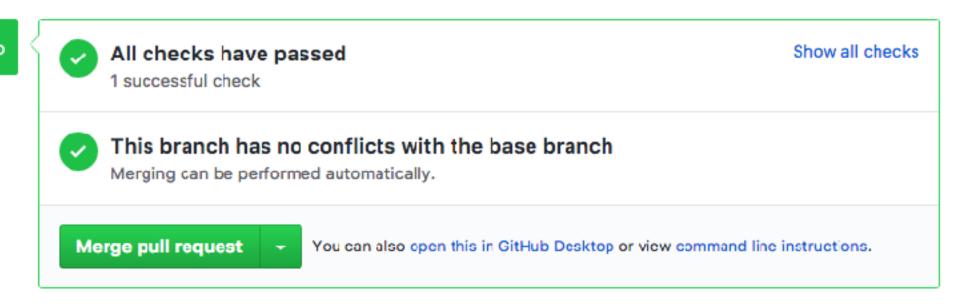
## Changelogs

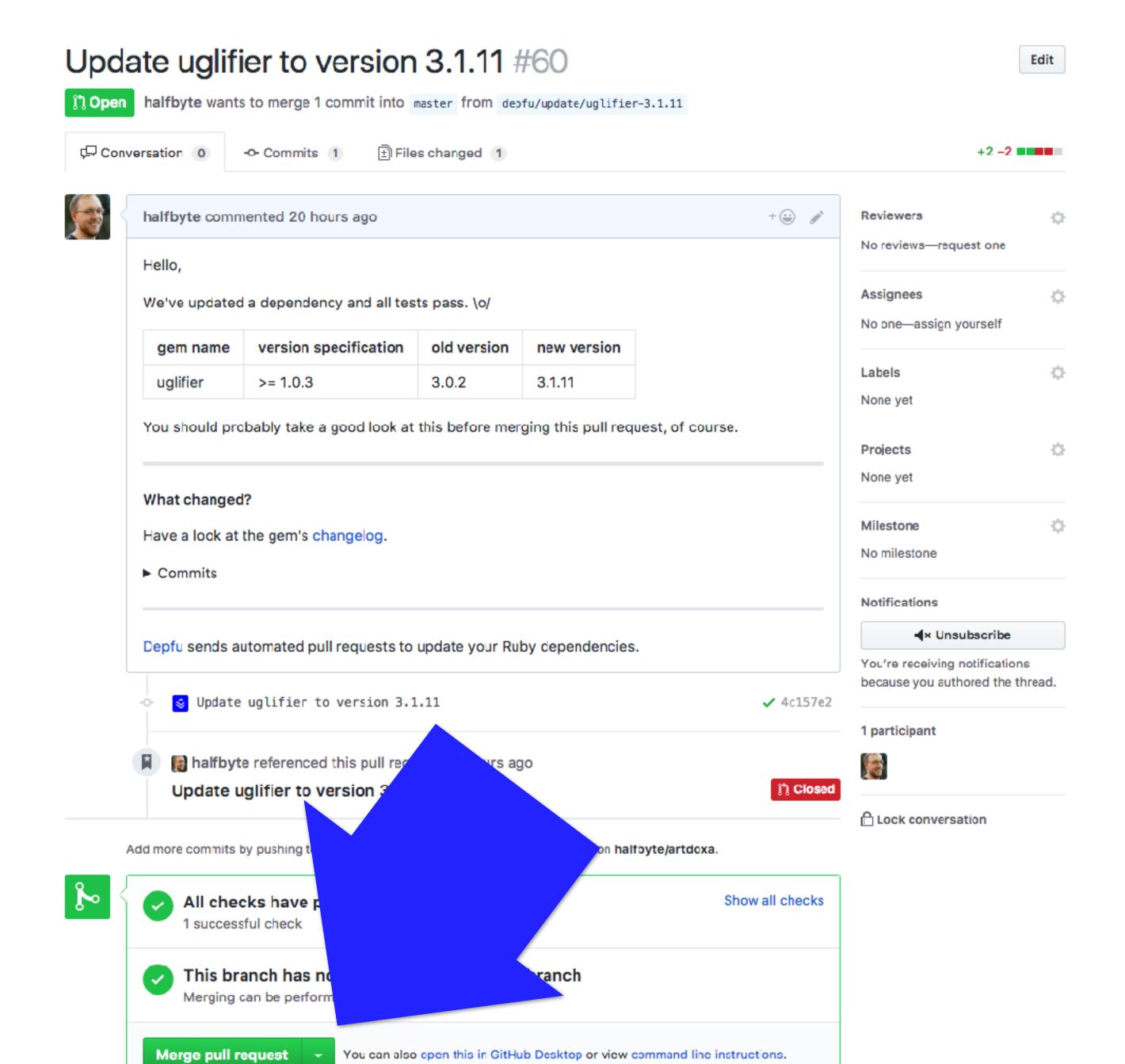
## Changelogs

## Changelogs



Add more commits by pushing to the depfu/update/uglifier-3.1.11 branch on halfbyte/artdoxa.





## Best practice

### Workflow

## Keep your software up to date

# Look at all the things we're not doing.

## Nobody wants another inbox!

## Pull requests: conversations





## Delete branches

## Close old duplicates

## Keep noise down

### Lesson #7

Good automations stay out of the way.

## Why not automate?

## Job security

# Ain't nobody got time for that

### Loss of control

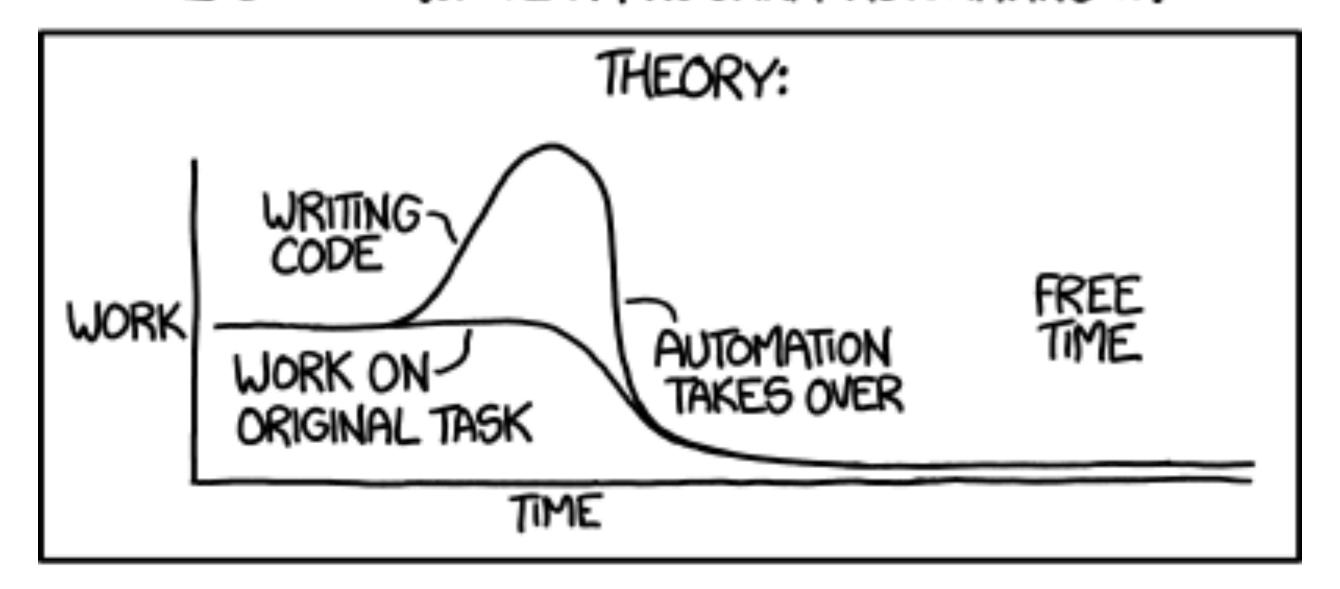
## Feeling?

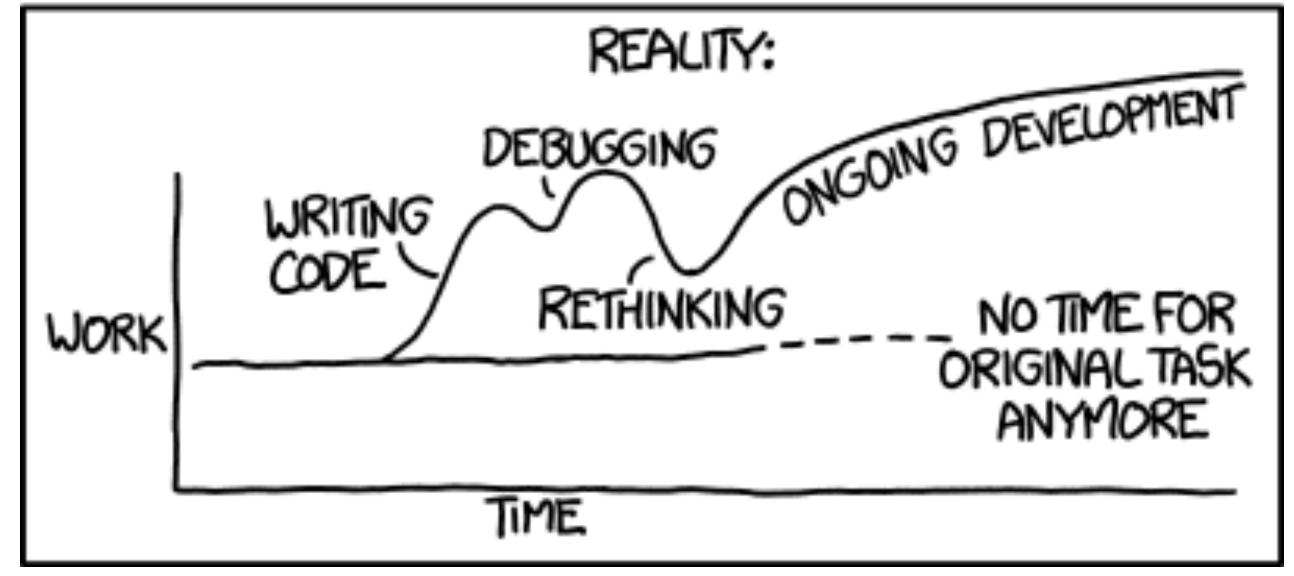
## SCALE

### scale

### Cost vs. benefit

#### "I SPEND A LOT OF TIME ON THIS TASK. I SHOULD WRITE A PROGRAM AUTOMATING IT!"





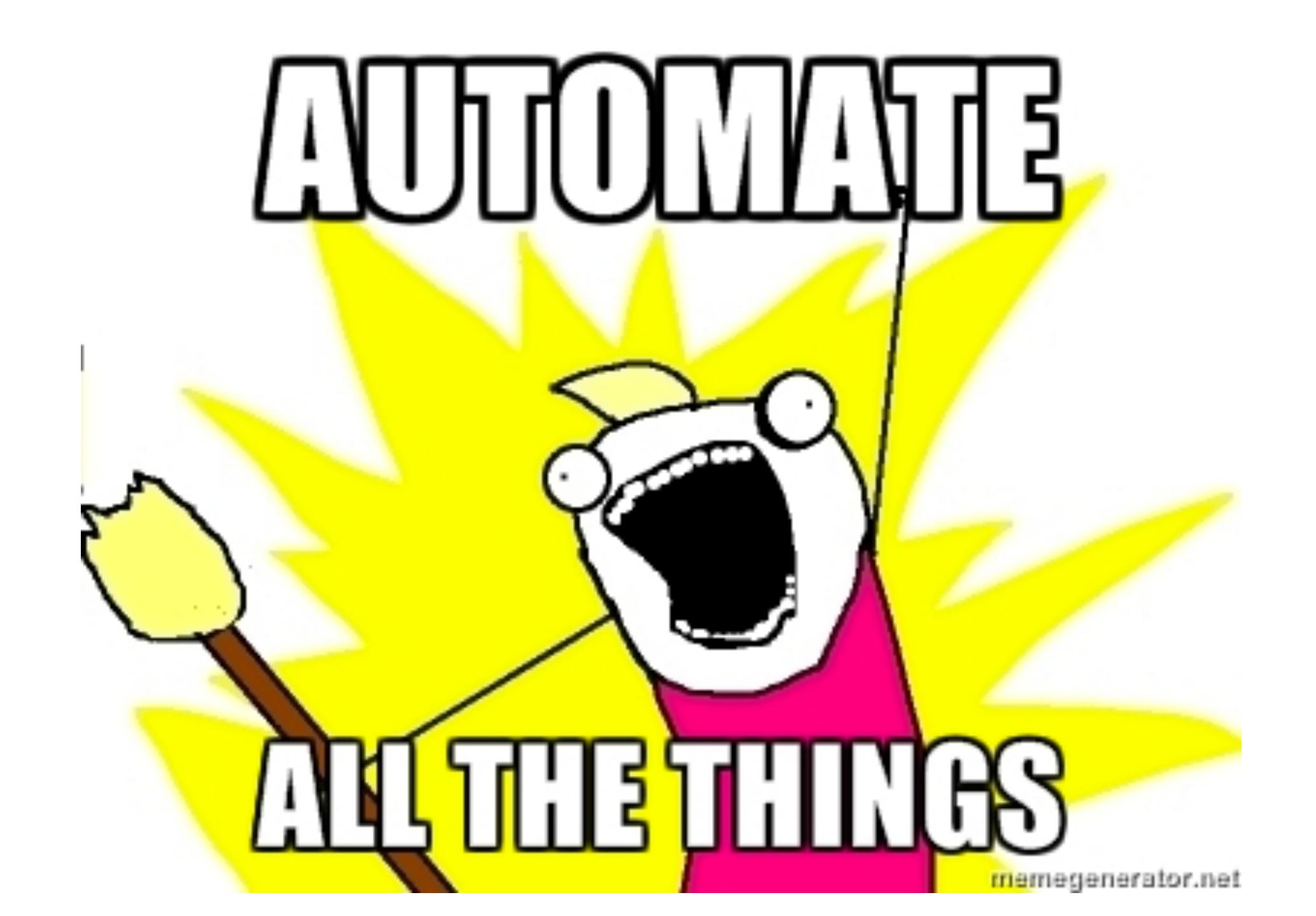
### Risk vs. benefit

#### Lesson #6.1

Automating high risk tasks almost always only are worth it on large scale systems.

# Intrinsic systems knowledge

#### Observation



#### Premature automation



#### More info

- John Allspaw on automation <u>part 1</u>, <u>part2</u>
- Courtney Nash's <u>automation myths</u>
- Of course: <u>depfu.io</u> <u>heydepfu</u>
- Me: ♠ halfbyte ★ halfbyte ★ <jan@depfu.io>
- THANK YOU SO MUCH! 💚 💚