



# Speeding up your testflow





## The multitasker



## The purist



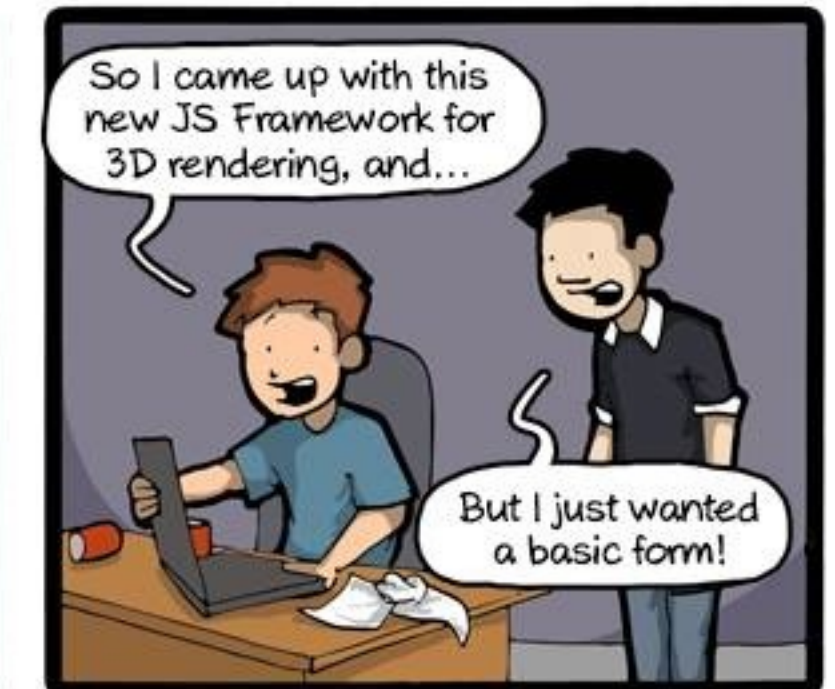
## The multitasker



## The over-focused guy



## The R&D addict



## The one who's never heard of Google



## The guy "working" from home





Speeding up ~~your~~ my testflow



# Goal

one continuous, smooth, line of thought



# The problem

- Rails load time is slow
- significantly slows me down when
  - doing TDD
  - breaking things to figure out how they work



# Solutions

- decouple from Rails (clever design)
- speed up Rails load time



# Zeus

- one master process written in Go
- relies heavily on *fork()*
- spawns slaves (instances of your app)
- transparently connects your terminal to a running slave





# Spork

- uses a distributed object system (dRuby)
- no add-in services needed beyond those provided by the Ruby runtime (e.g. TCP sockets)



# Spring

- implemented in pure Ruby
- uses the Rails built-in code reloader
- ships with Rails  $\geq 4.1$





DEMO

# Demo application

- Devise demo application
- 1 model, 3 controllers + unit tests
- 9 acceptance tests
- Rails 4.2, Bootstrap-Sass 3.3, RSpec 3.1, Capybara 2.4



# Why I'm moving away from Zeus

- sometimes messes up constant loading
- doesn't understand nesting of `shared_examples`
- Spring comes bundled with Rails  $\geq 4.1$

# Conclusion

- constant tweaking of your workflow will make development faster
- play around & experiment with tools, it's really rewarding