devops with docker

@beshippable

avi cavale

co-founder & ceo

@avinci || avi@shippable.com

shippable

continuous app delivery pipelines

the most frequently asked question
during an outage

the most frequently asked question
during an outage

what changed?

the immediate next question is….

the immediate next question is….

who made the change?

both questions are bad for your business

both questions are bad for your business

slowing down

both questions are bad for your business

slowing down

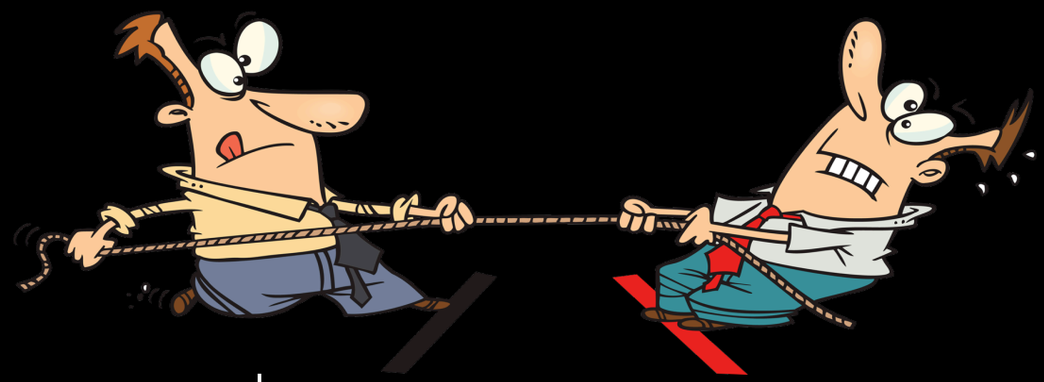
finger pointing

courtesy : dilbert comics

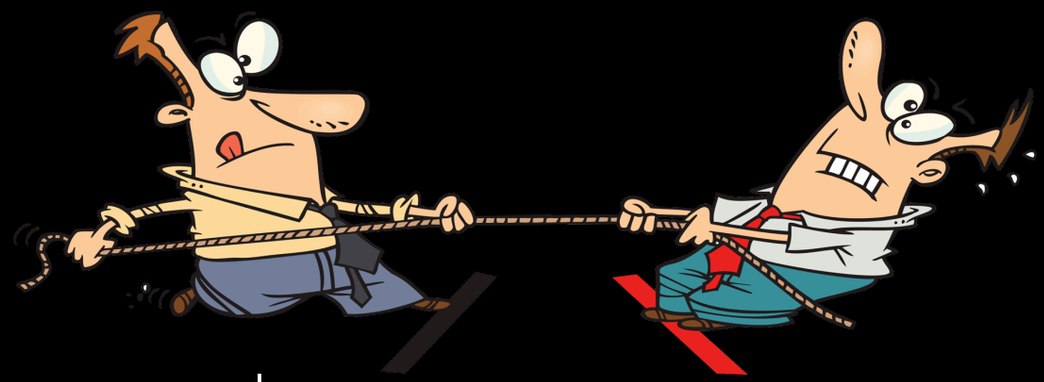
change is the root cause of almost all outages, but…

change is the root cause of almost all outages, but…

“business requires change”



“need to embrace change”



“need to embrace change”

devops: a CULTURE of embracing change

we embraced it tightly

traditional

application delivery pipeline

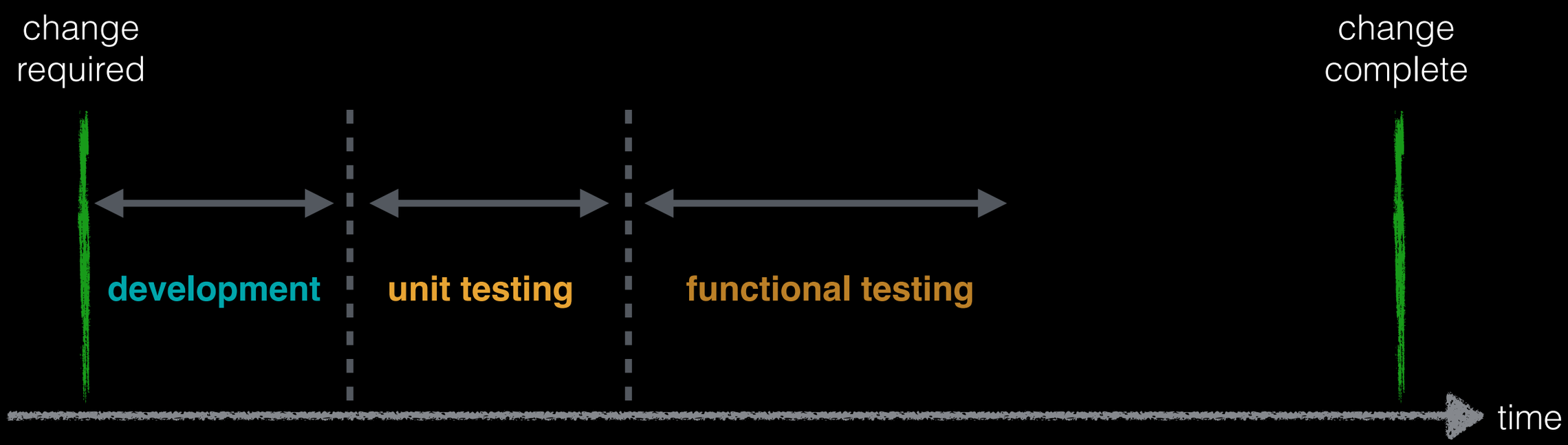
traditional

application delivery pipeline

traditional

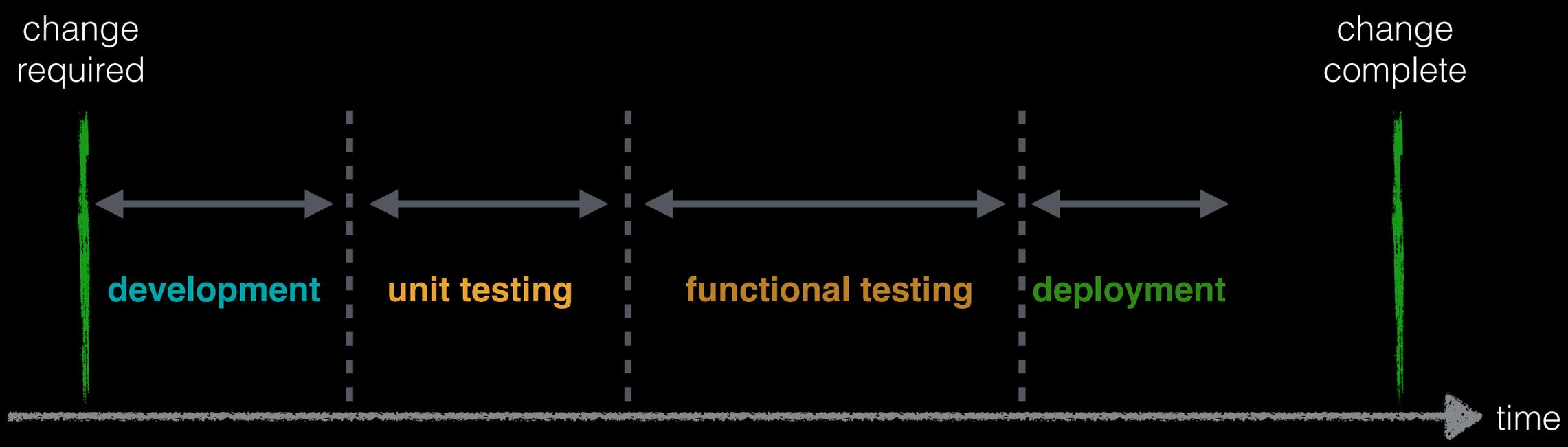
application delivery pipeline

traditional



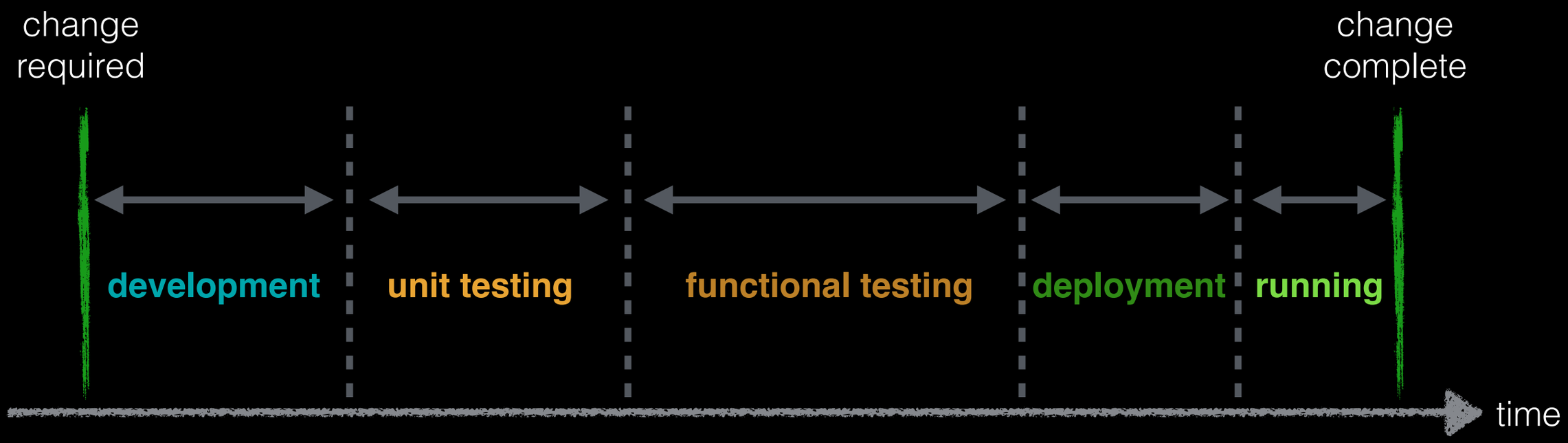
application delivery pipeline

traditional



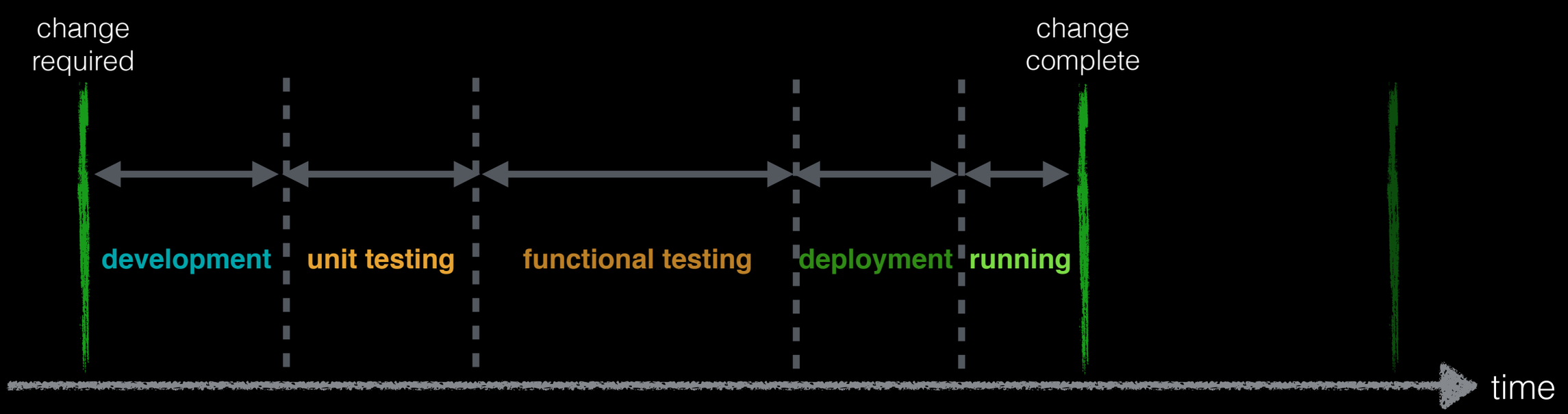
application delivery pipeline

traditional



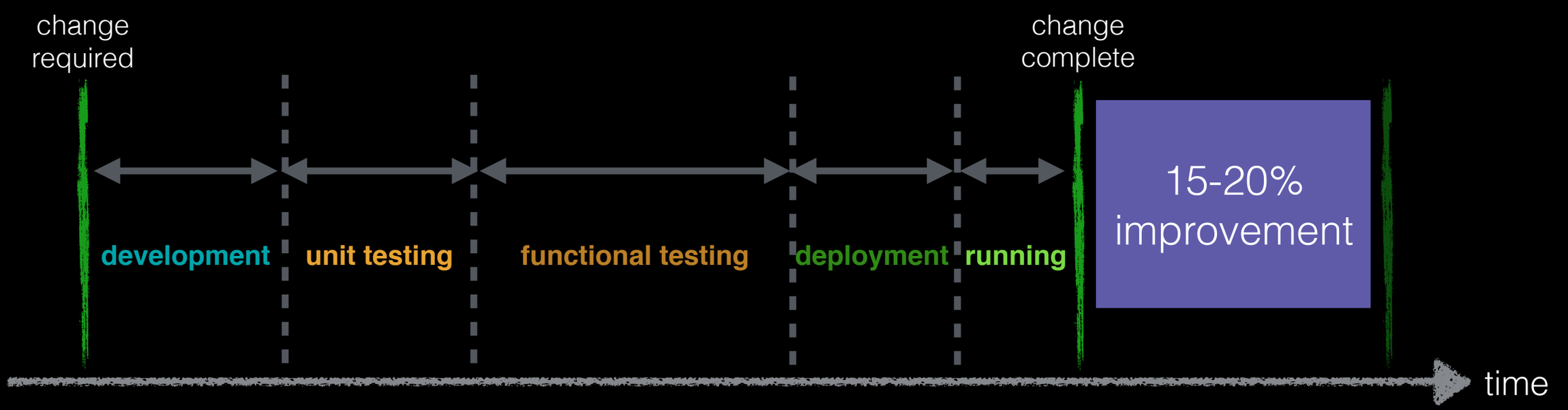
application delivery pipeline

devops culture embraced



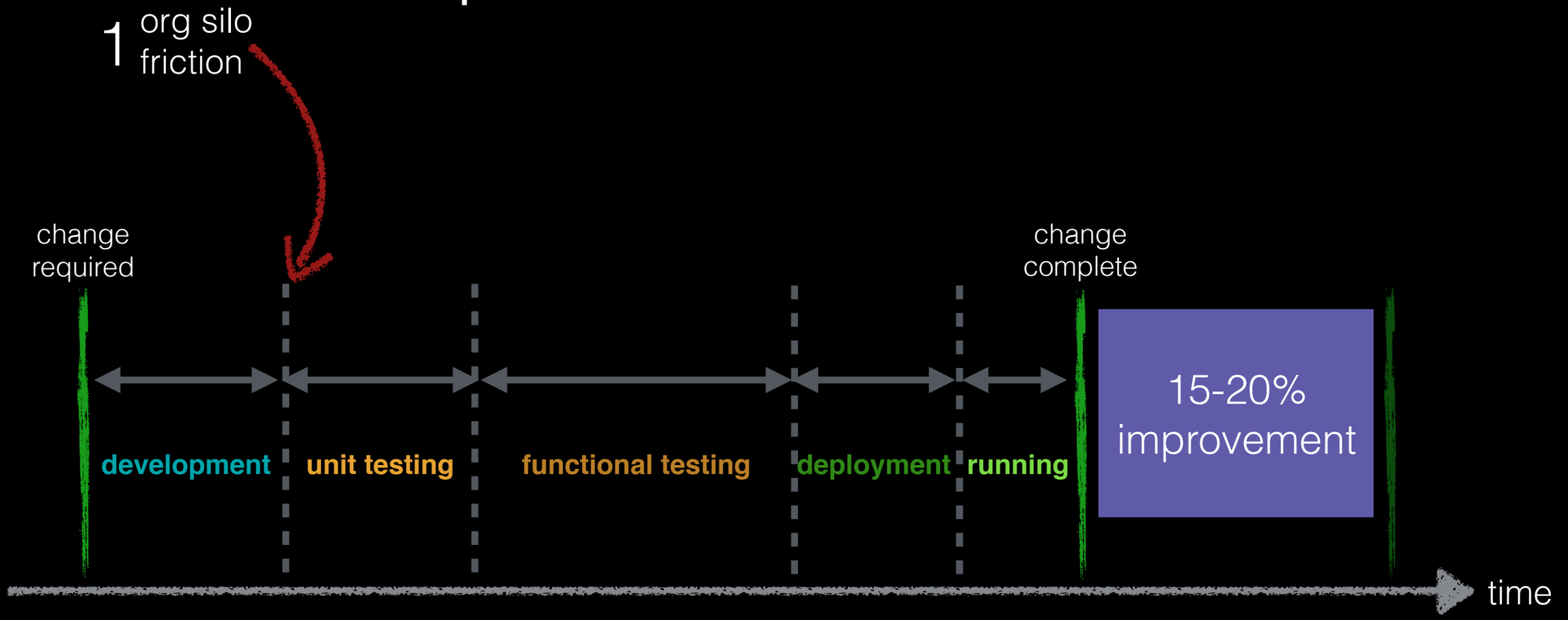
application delivery pipeline

devops culture embraced



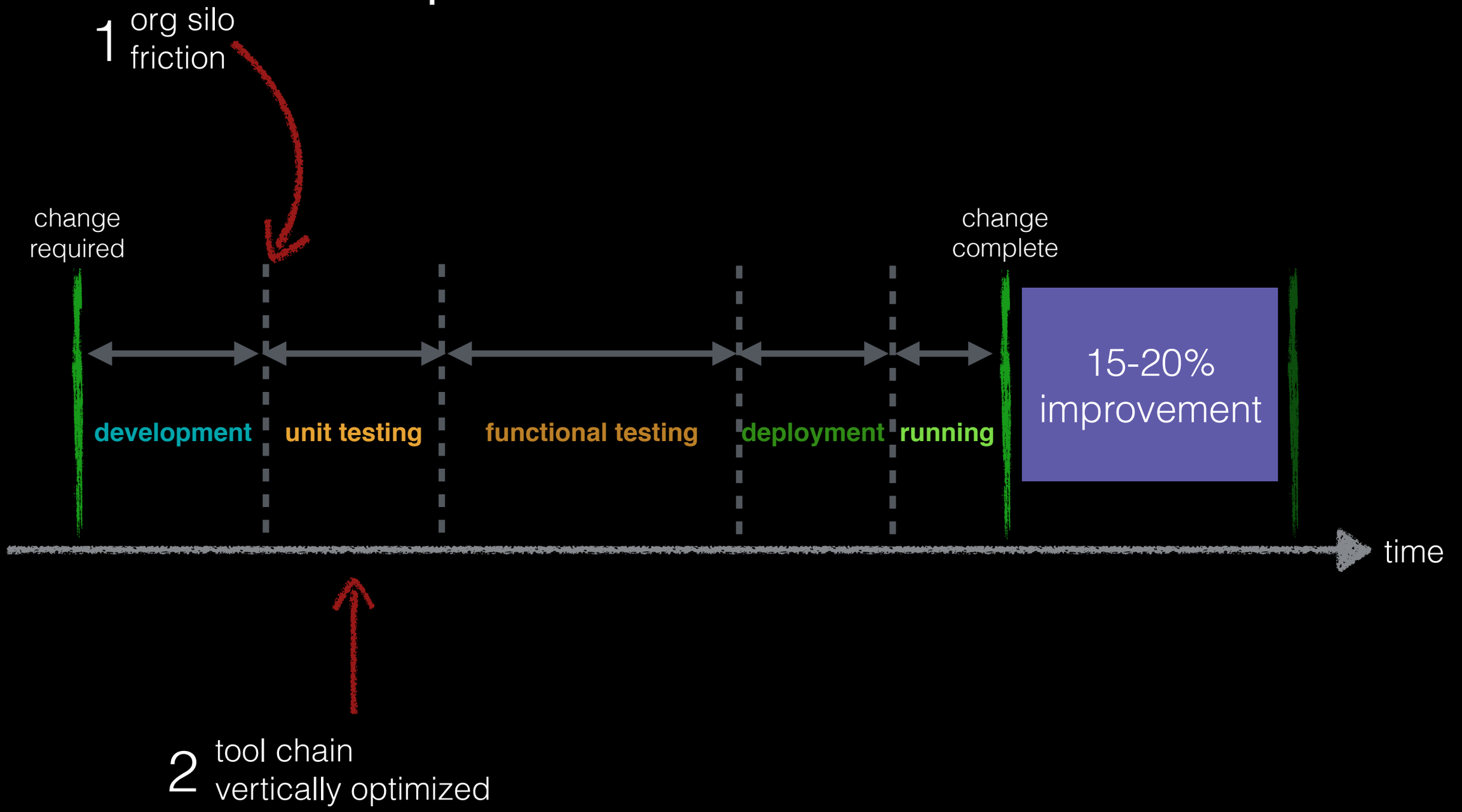
application delivery pipeline

devops culture embraced



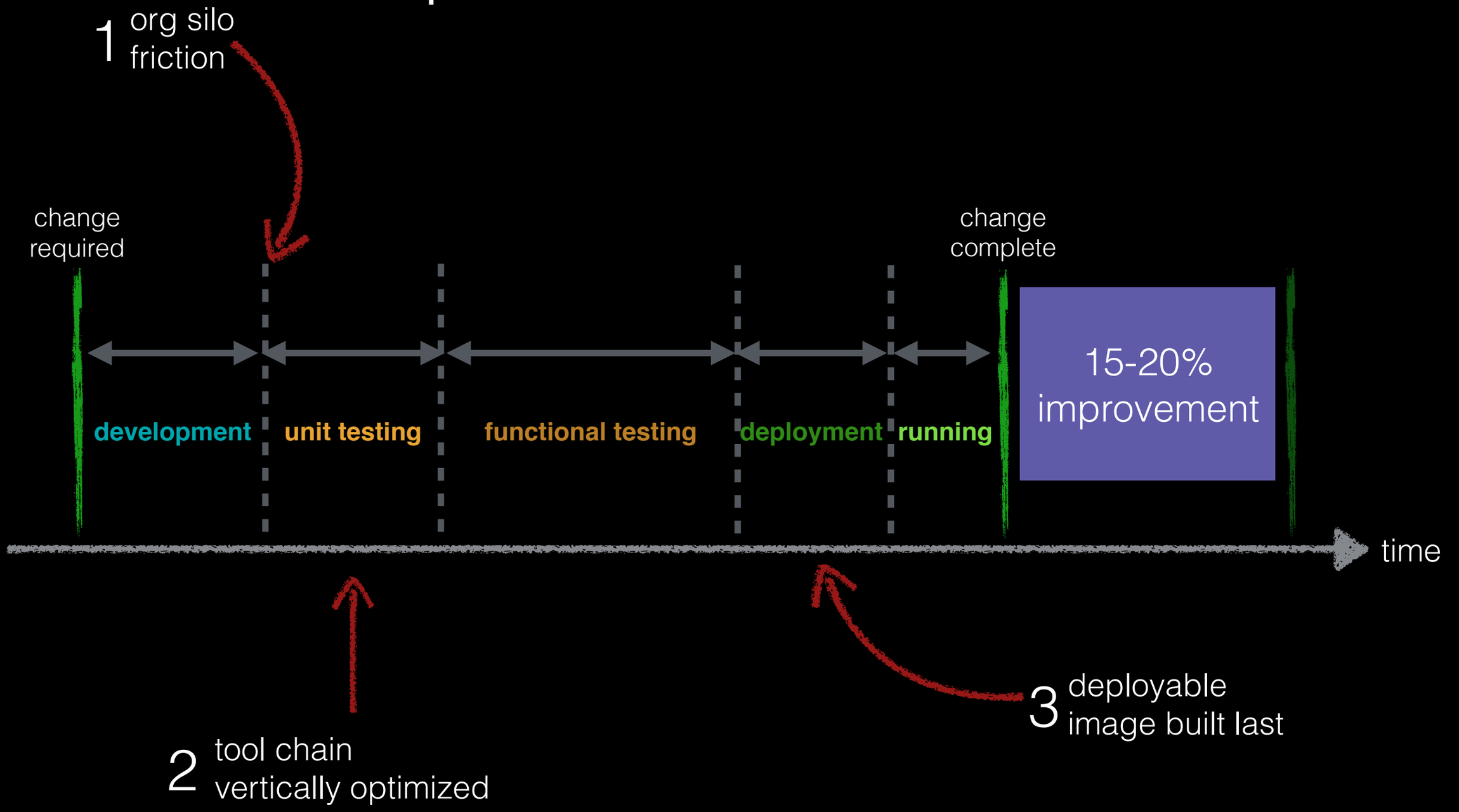
application delivery pipeline

devops culture embraced



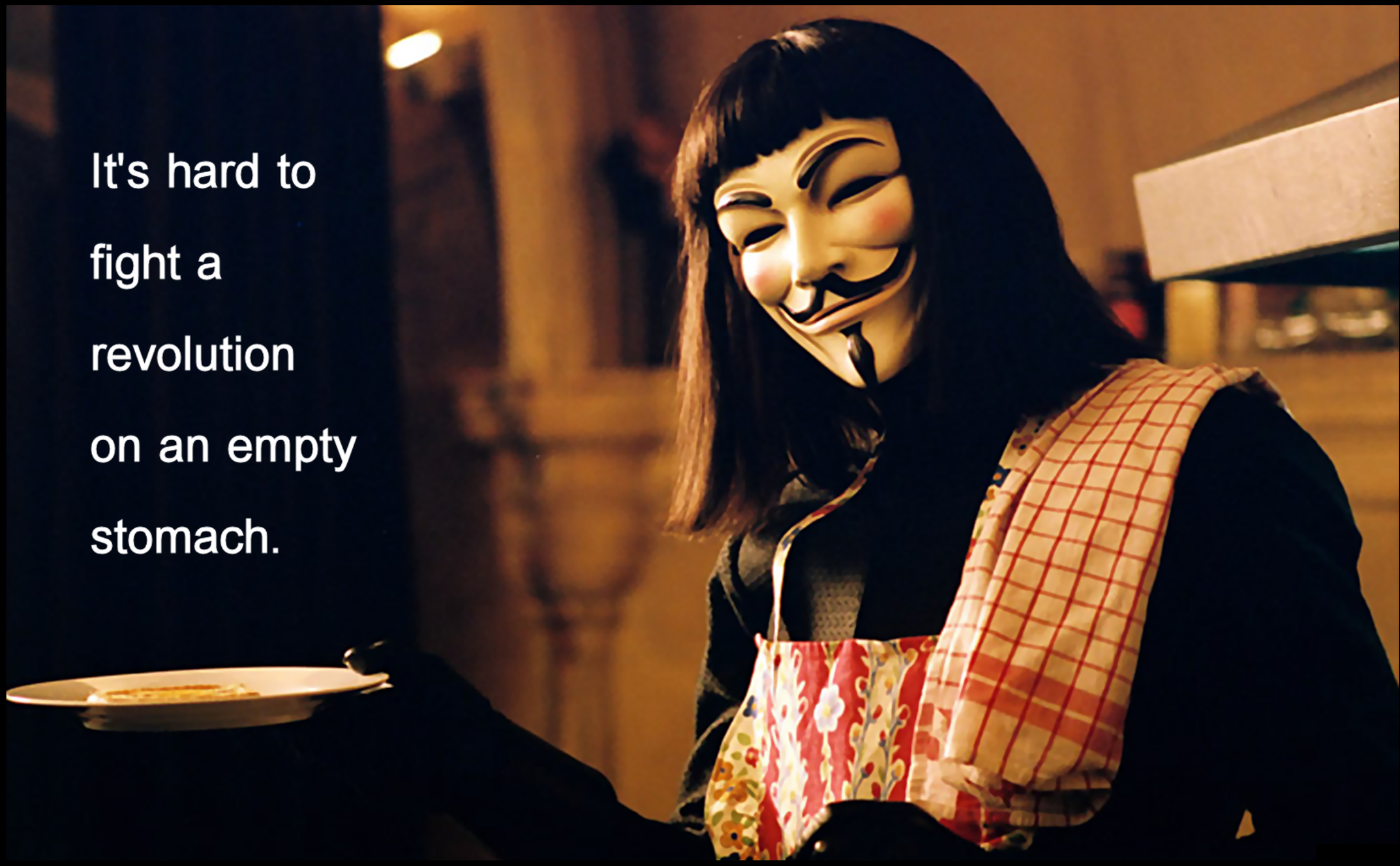
application delivery pipeline

devops culture embraced



application delivery pipeline

20% better is not a REVOLUTION!



courtesy : V for Vendetta

why not build deployable image first?

courtesy : V for Vendetta

devops is culture, so to make it a

F

A

C

T

devops is culture, so to make it a

F rictionless

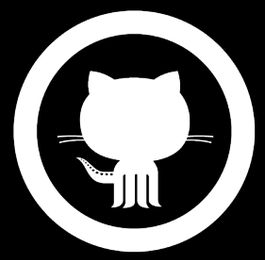
A gile

C ontinuous

T ransparent

application delivery pipelines

if you want to do just 1 thing this year



Transparency

every action

- starts in a chat room
- tracked as a github issue
- ends in a chat room

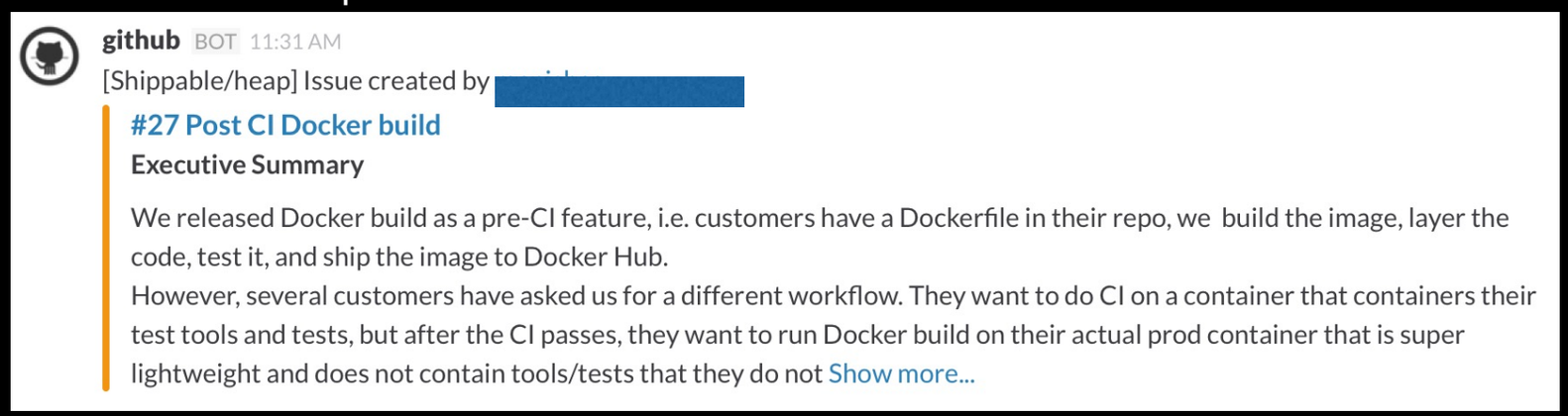
if you want to do just 1 thing this year

Transparency

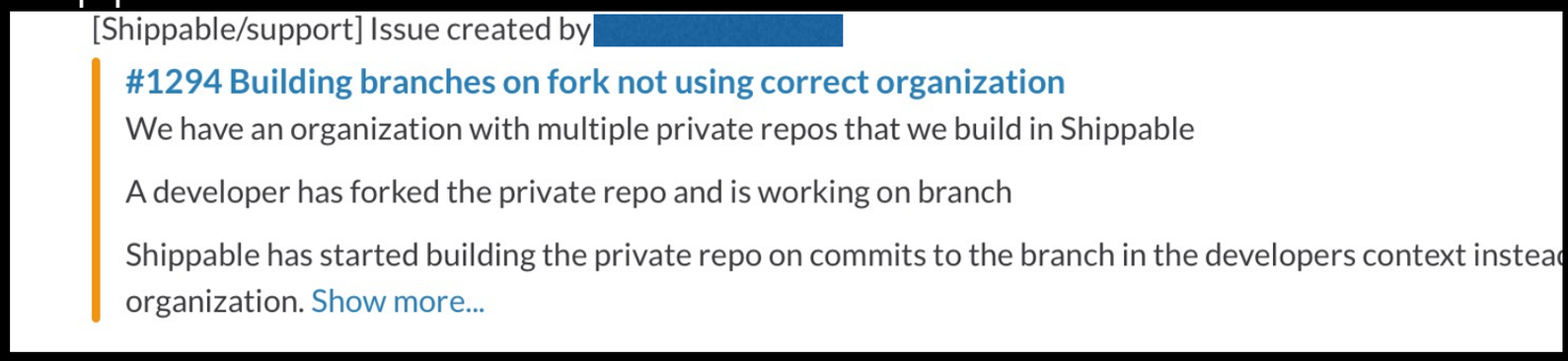
every action

- starts in a chat room
- tracked as a github issue
- ends in a chat room

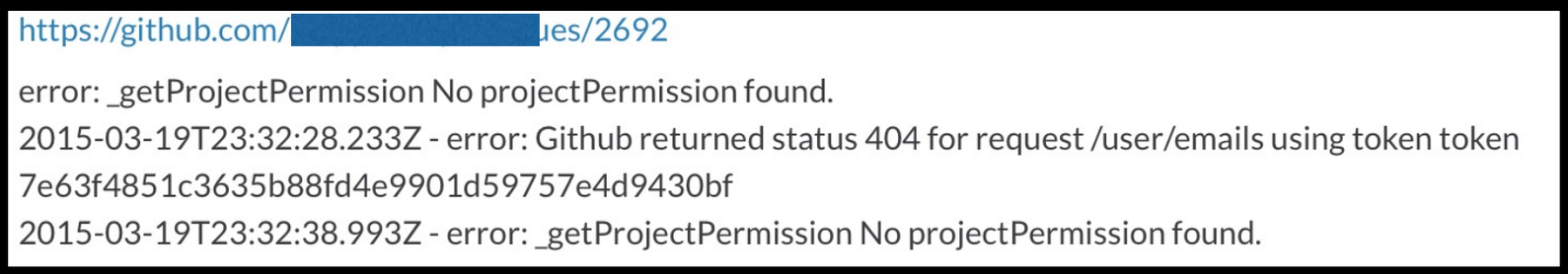
business req



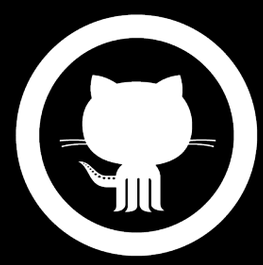
support



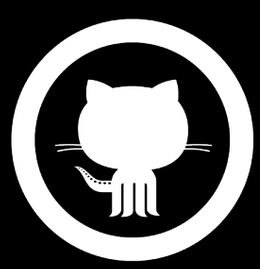
prod errors



Continuous



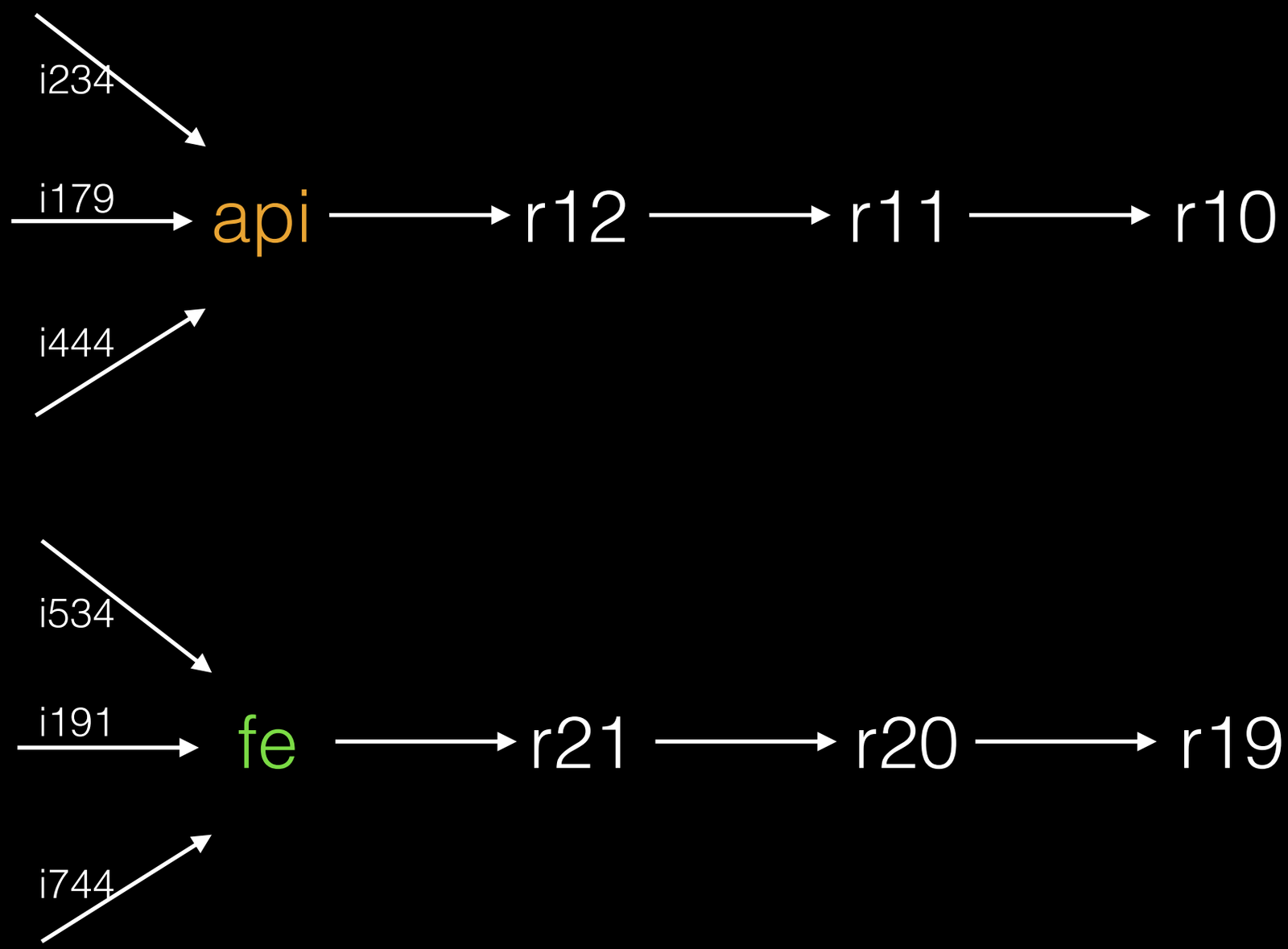
- continuous integration for every pull-request
- 1 step docker build image for every merge to `master`
- 1 step auto deploy & continuous functional testing
- deployment to prod is just changing an ENV var

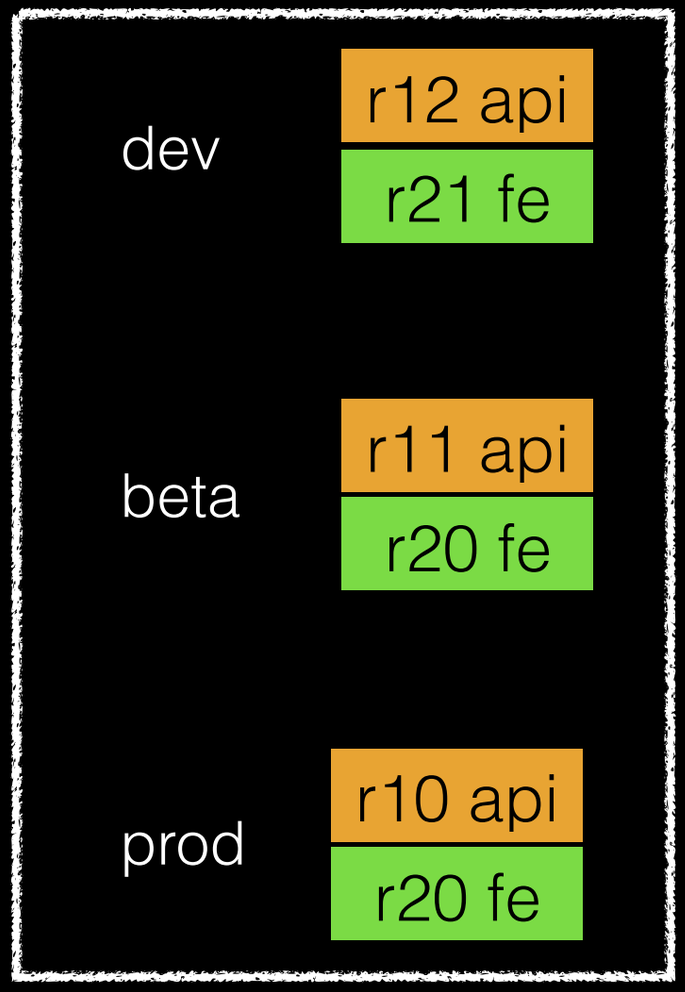


Agile

- no SSH!
  - bug fixes, new changes, ops scripts all go the same way
- always deployable `master`
  - no feature branching
- feature flags
  - in-code branching
- small frequent changes
  - beta program

Frictionless





app delivery pipelines

container fab

noticeable FACTs

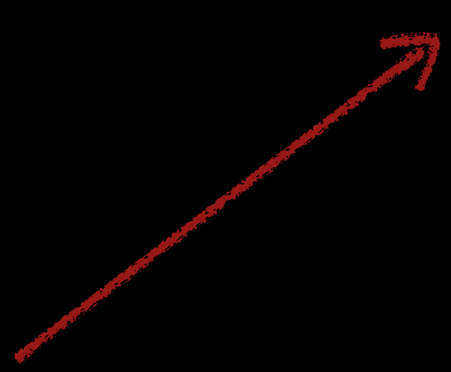
ship images 40-60 times a day

15 devs in 5 timezones

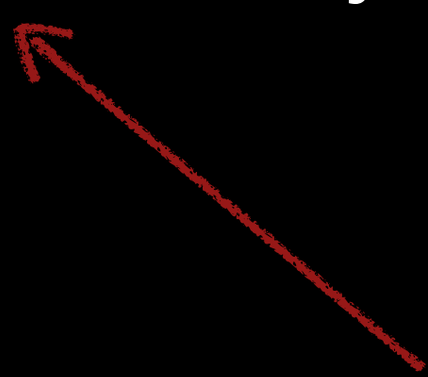
2hrs mean time to resolve bugs

where are we going from here?

micro-services baby!



throughput creates
merge hell



rapid changes
on 1 repo creates
images chaos

25,000 build
containers/week

45,000 devs
on Shippable

signup for our beta program

email me avi@shippable.com

thank you very much

questions?

@beshippable