Continuous Delivery
Antipatterns

JavaOne 2015
Andrzej Grzesik

andrzej grzesik

@ags313

andrzej@grzesik.it

andrzejgrzesik.info

polish java user group

GeeCON

[sckrk]

cracow hadoop group

my opinions are my own

disclaimer

@JavaOneConf @ags313 #JavaONE

please tweet!

questions?

just ask!

tl; dr:

release more often!

software

is a people problem

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

agile manifesto, 2001

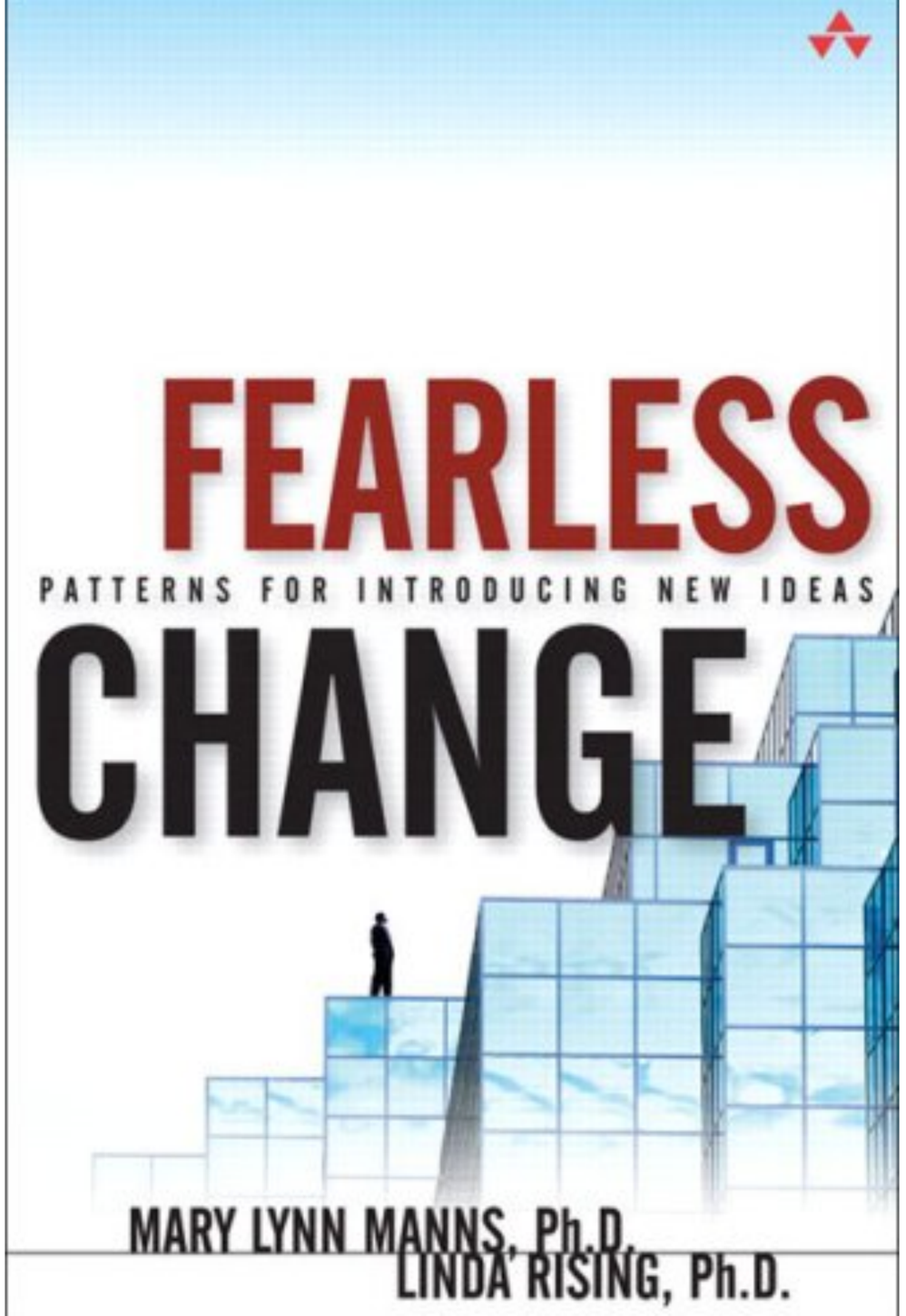
delivery is
organization-specific

change scenario

1. prioritize problems

2. fix first

repeat


FEARLESS
PATTERNS FOR INTRODUCING NEW IDEAS
CHANGE
MARY LYNN MANNS, Ph.D.
LINDA RISING, Ph.D.

when system breaks

money is lost

policies as signs of failures past

control the unpleasant

or at least try

fear of releasing

(fear it’s not technical)

fear of things breaking

how to fight fear?

build trust

gently!

foreign customer, shaky component

send unit tests with code!

and wait

symptoms?

planned stagnation!

quiet periods

release ‘trains’

rc, beta, gold

solution: **involve** business

a.k.a. ‘manage stakeholders’

small changes
should happen quickly

great selling point

when system breaks

all roll-back!

to where? from where?

problem:
no idea what is where

git push --force prod

#randomhashisbetterthannone

do: --version

better:
do semantic versioning

http://semver.org/

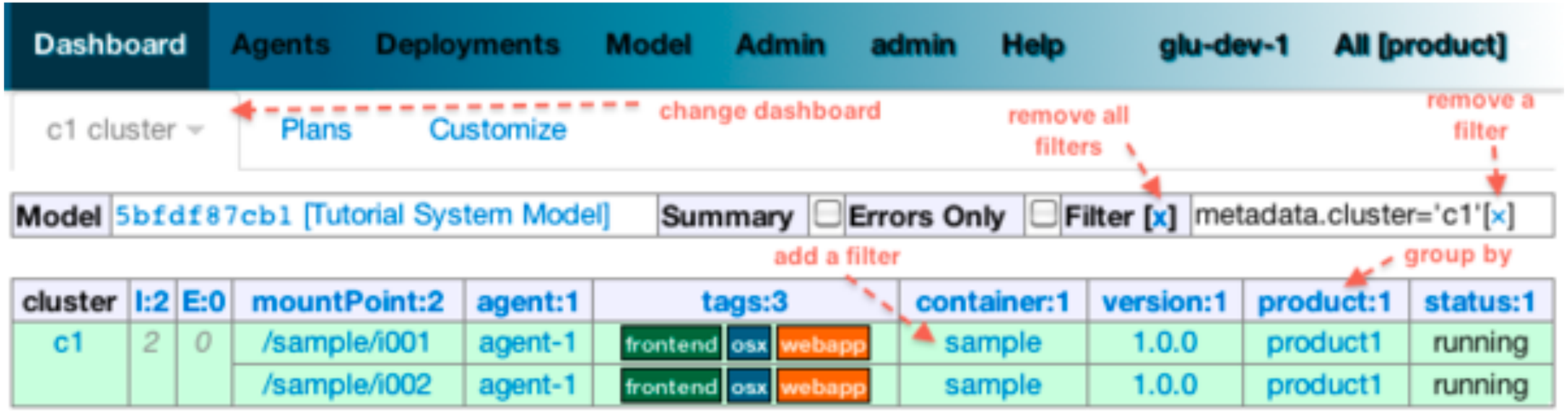
do: know what is where

do: know what is where

(have a dashboard)

glu

https://github.com/pongasoft/glu



versions vs frontend

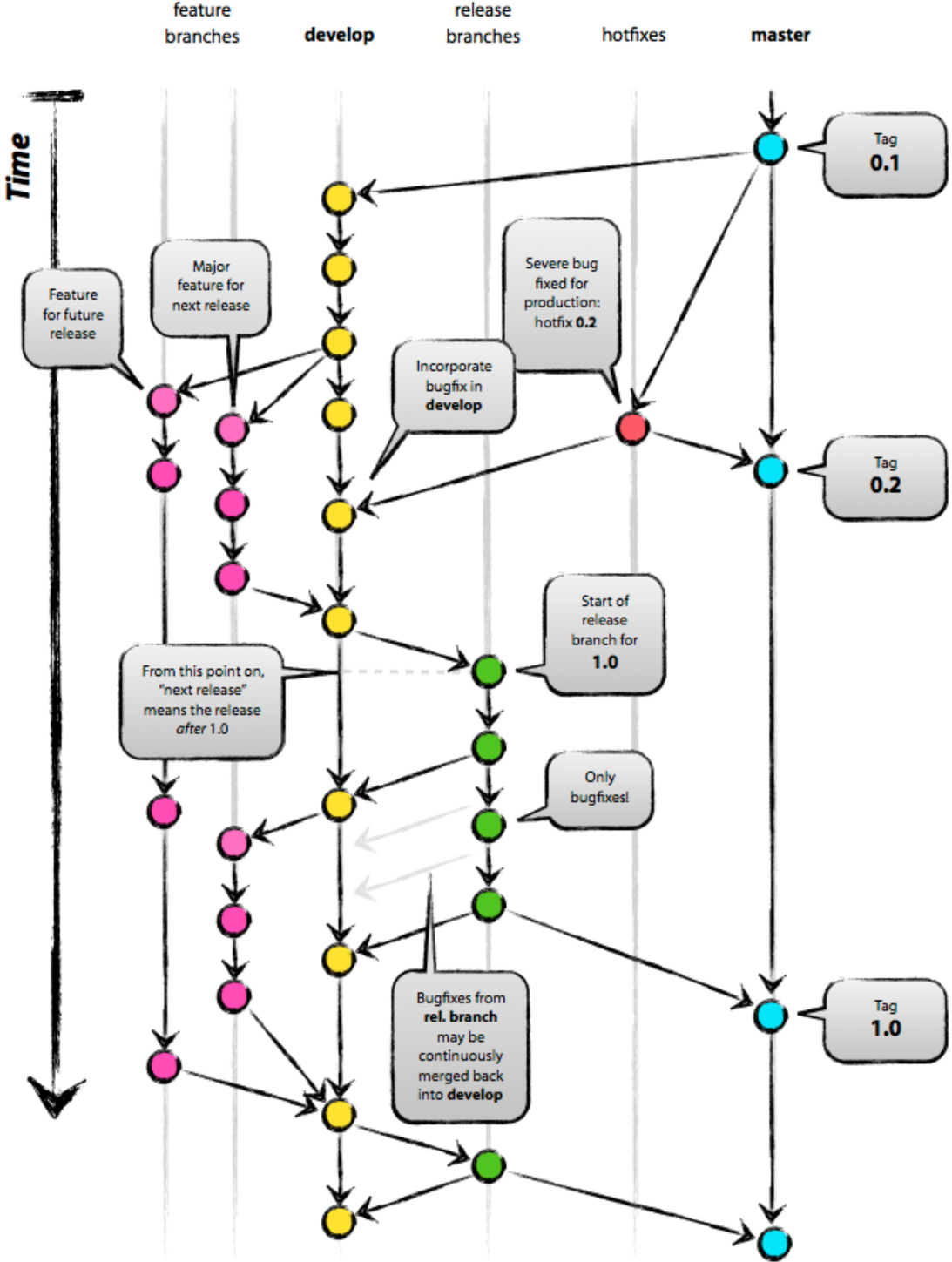
problem:
many teams/unstable code

multiple repo is OK

one for dev, one for releases

git flow is OK

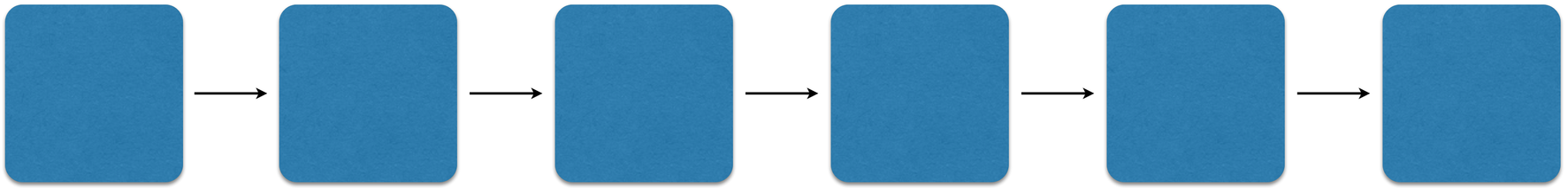
http://nvie.com/posts/a-successful-git-branching-model/



personal favourite:
stable master

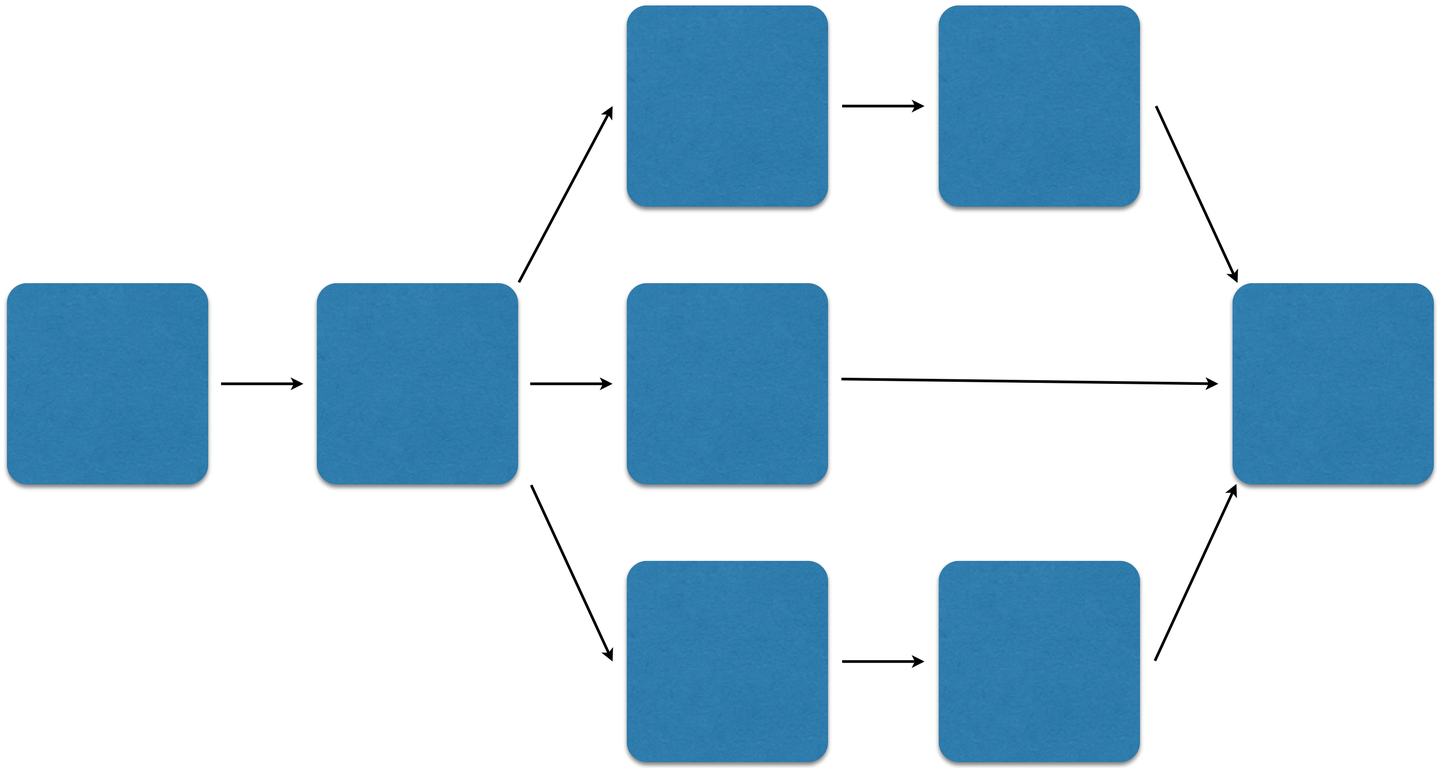
Build && Environments

ok, let’s have a pipeline

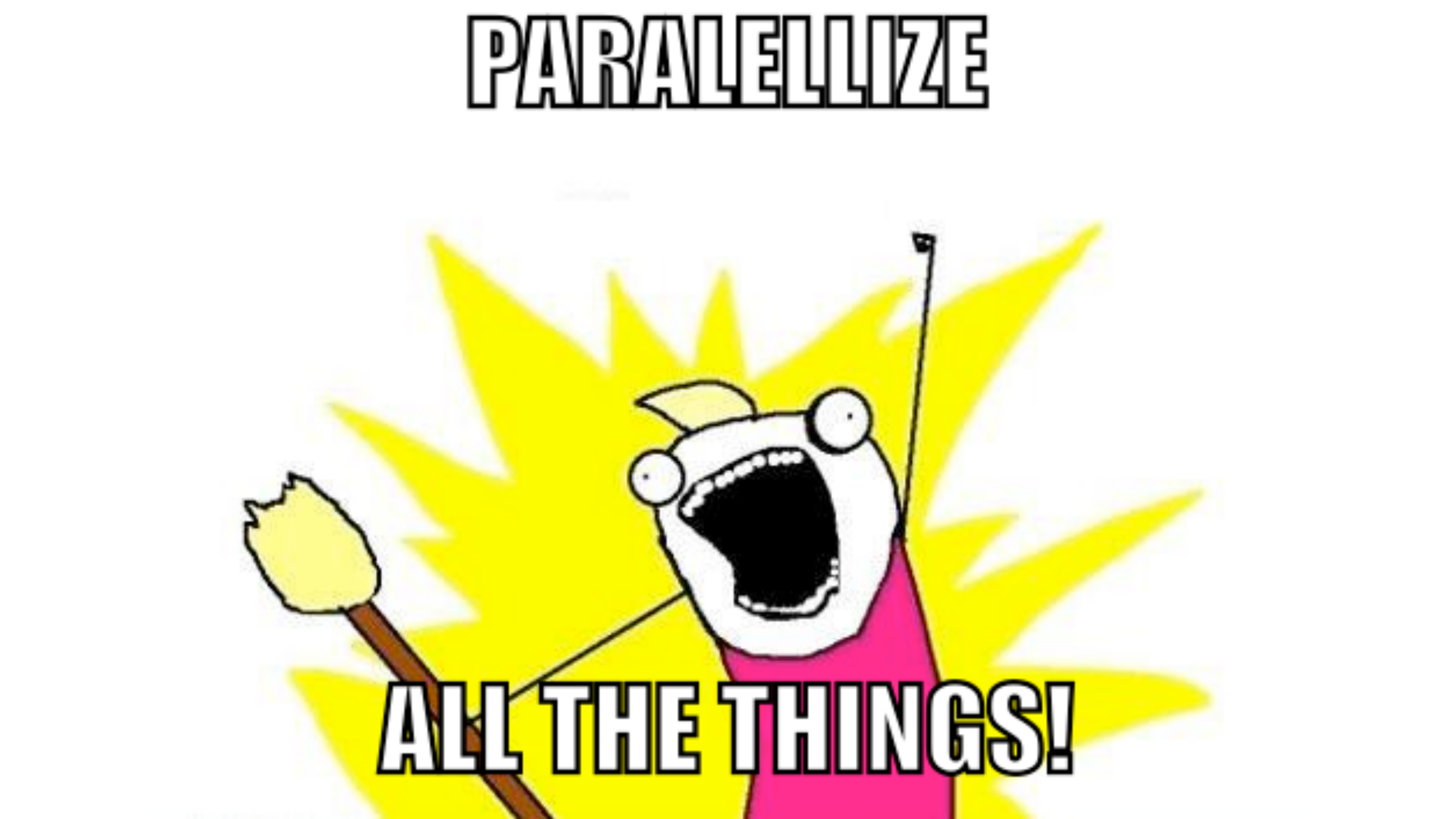


problem:
slow

solution:
paralellize



which part?

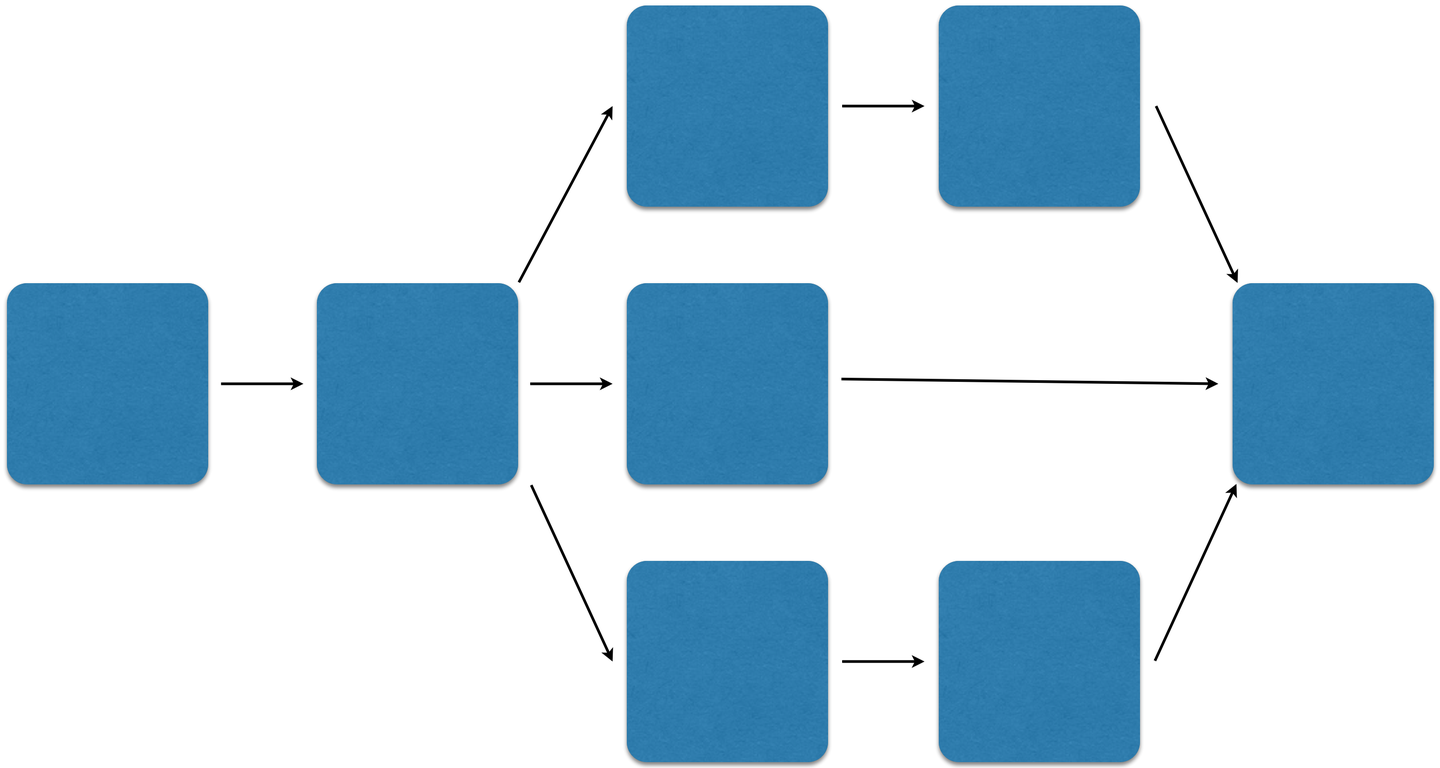

PARALELLIZE
ALL THE THINGS!

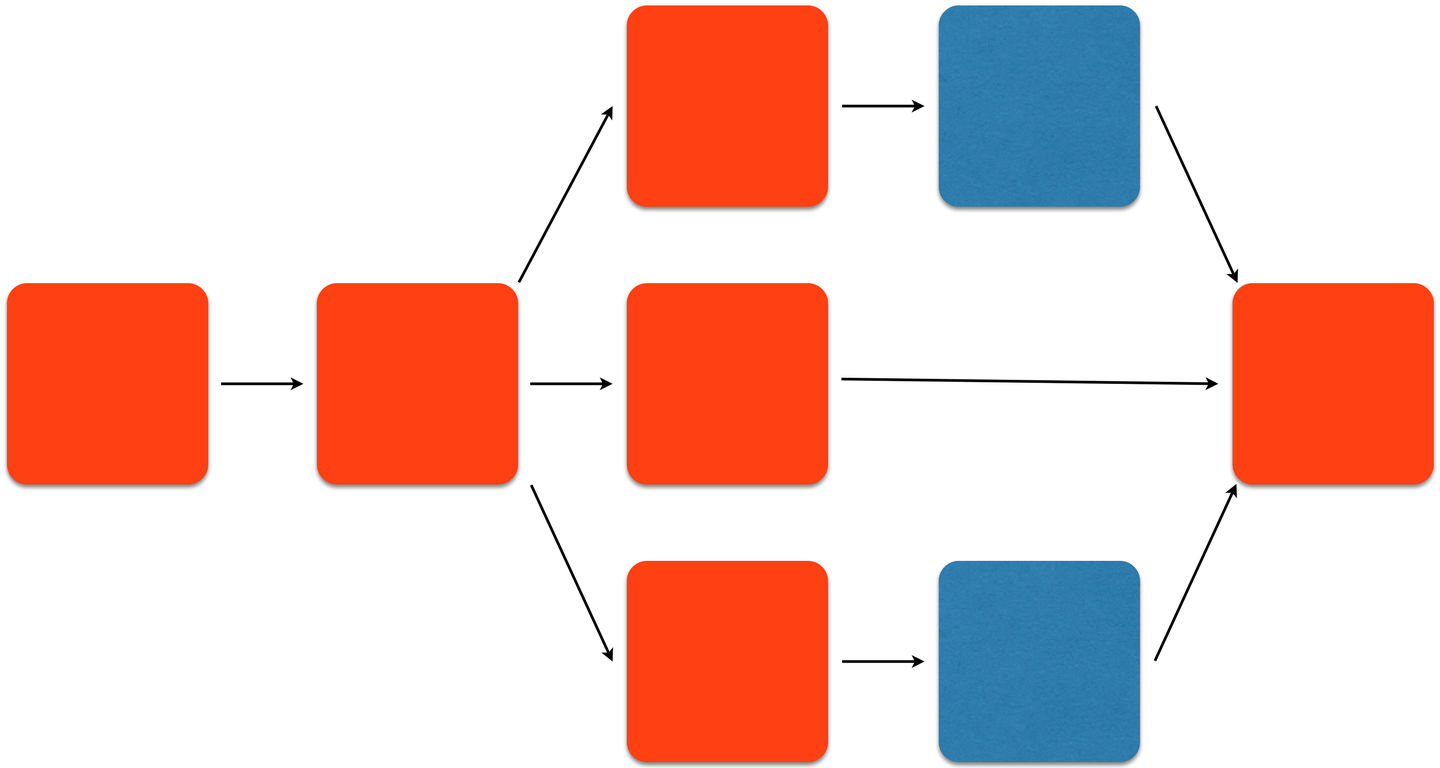
free:

more decoupled design

problem: **re**compiling

Implementing a Fibonacci

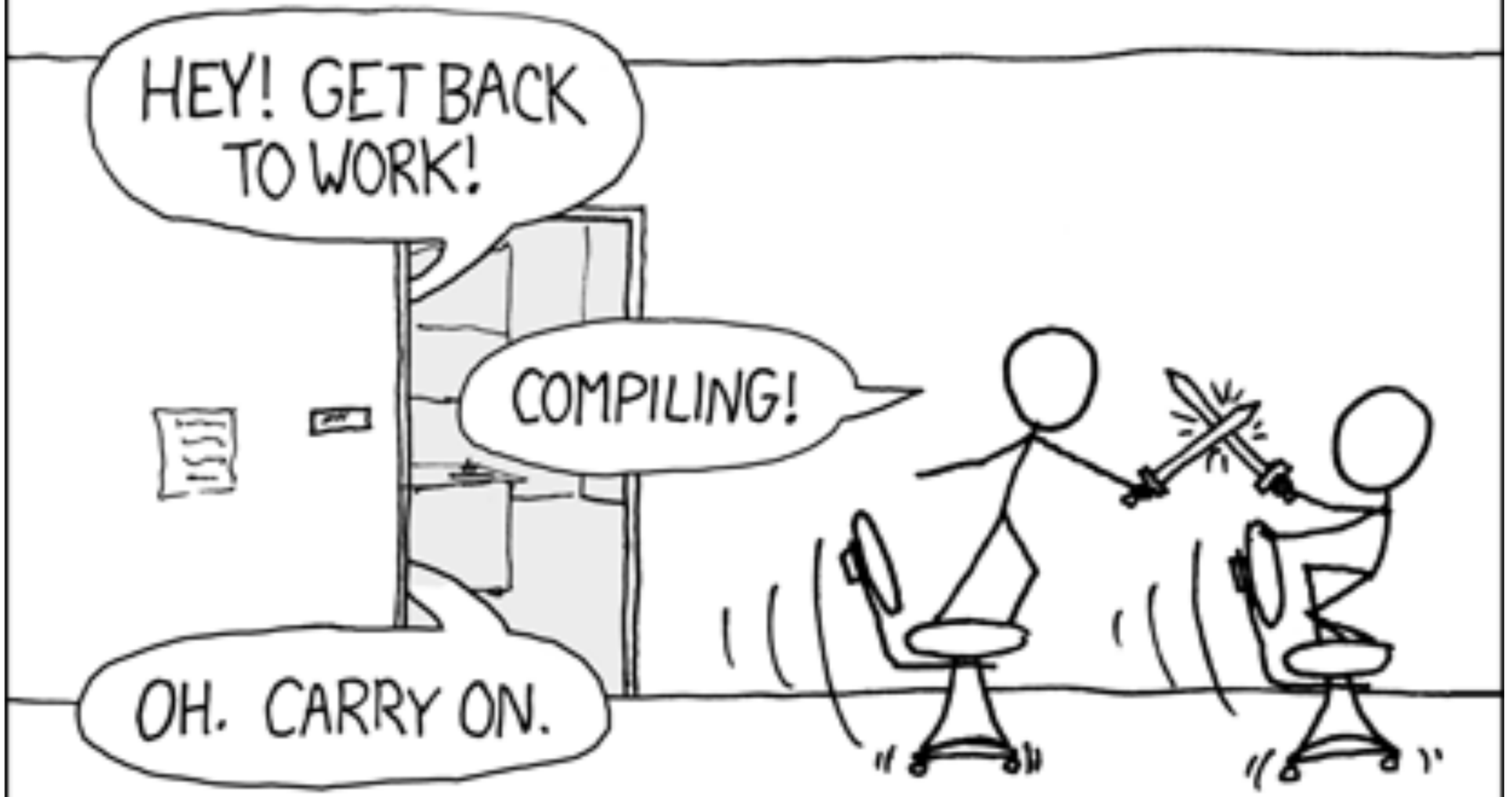




relevant to VMs, containers etc

THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."



how to replicate production?

do: use a binary repository

What about Docker?

save intermediary states

problem:
separate teams

release team

dealing with ‘**danger**‘

symptom:
dealing with **danger**

likes to be manual

‘job security‘

us vs them

‘leave me alone, I’m important’

release processes

curious release processes

Work expands so as to fill the time available for its completion

Parkinson’s Law

too curious processes
lead to

unofficial releases

unofficial releases

(don’t do them)

Bunkers

solution:
encourage **interactions**

break && integrate

break && integrate

repeatedly

games are awesome!

repeat the event

problem:
manual infrastructure

infrastructure

we use chef, we're safe

^^

automate everything!

Puppet vs Chef vs Ansible vs…

does not matter

did you **test**?

Bring Behavior-Driven Development to Infrastructure as Code

Test-Driven Infrastructure with Chef

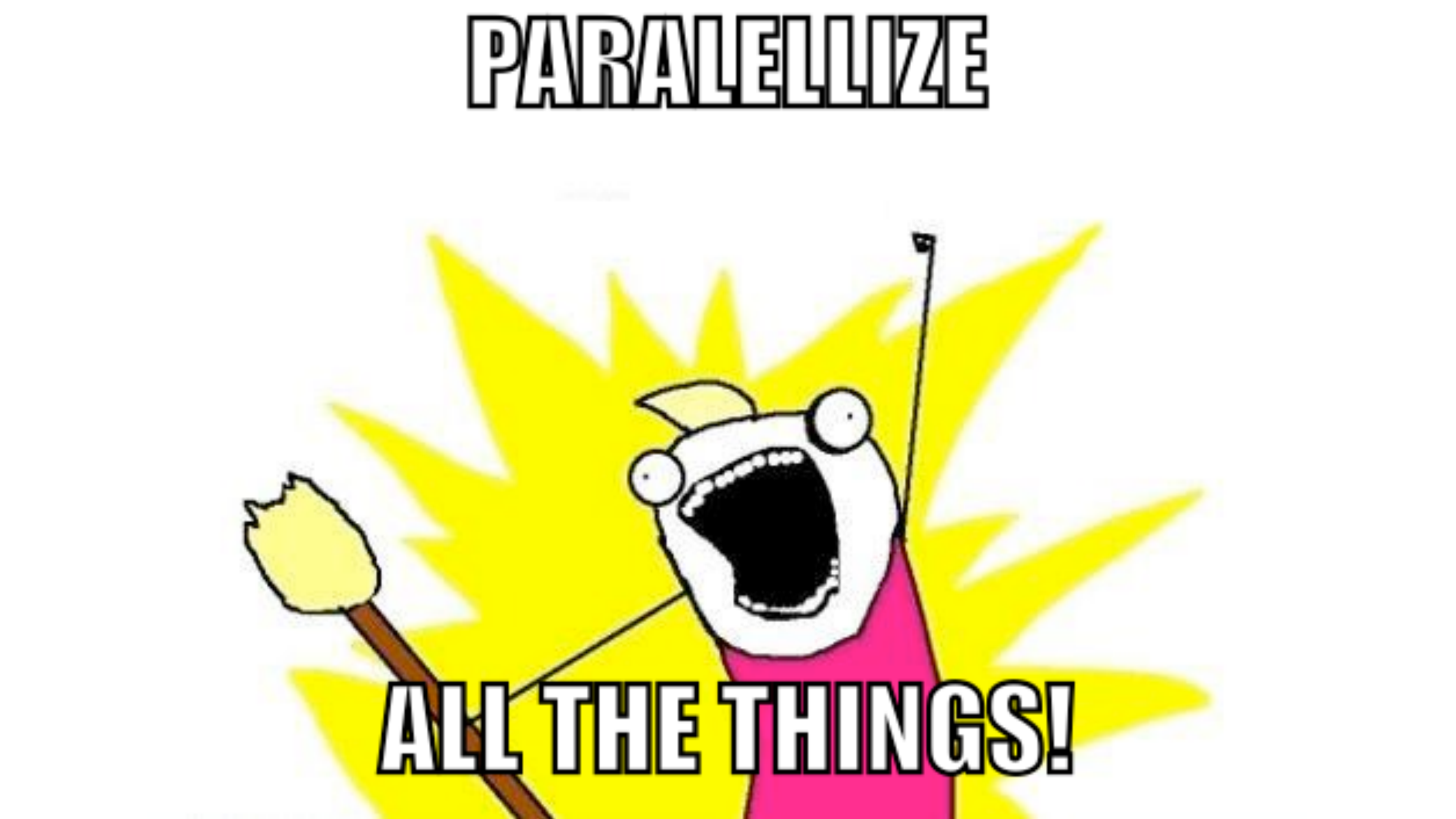
O'REILLY®

Stephen Nelson-Smith

developers doing infrastructure

problem:
Env builds

slow?


PARALELLIZE
ALL THE THINGS!

problem:
internet

quiz


TECHNIC

need internet to build?

of course!


O RLY?

rubygems.org
cpan.org
maven.org

rubygems.org

cpan.org

maven.org

go down!

run your build without the Internet

or at least try, you'll learn fun things :-)

application
and environment

save time and nerves

binary repo/proxy/…

problem:
no **runtime** upgrade

Release	Release Date
Java SE 7[98]	2011-07-28
Java SE 7 Update 1[99]	2011-10-18
Java SE 7 Update 2[100]	2011-12-12
Java SE 7 Update 3[101]	2012-02-14
Java SE 7 Update 4[103]	2012-04-26
Java SE 7 Update 5[104]	2012-06-12
Java SE 7 Update 6[106]	2012-08-14
Java SE 7 Update 7[108]	2012-08-30
Java SE 7 Update 9[109]	2012-10-16
Java SE 7 Update 10[111]	2012-12-11
Java SE 7 Update 11[112]	2013-01-13
Java SE 7 Update 13[114]	2013-02-01
Java SE 7 Update 15[115]	2013-02-19
Java SE 7 Update 17[116]	2013-03-04
Java SE 7 Update 21[117]	2013-04-16

mobile apps

webview is nice

frequent releases

make your user curious

make users say **bye**

change backend
you can

ask if new
features they want

forcing doesn’t work

problem:
deployment **failures**


2
3
DAYS WITHOUT
ACCIDEN

DEPLOYMENTS

DO YOU TRACK THEM?

FAILED DEPLOYMENT PROCEDURE

ROLLBACK? OR DOWNTIME?

do: **test** your rollback

as you test your backups

problem:

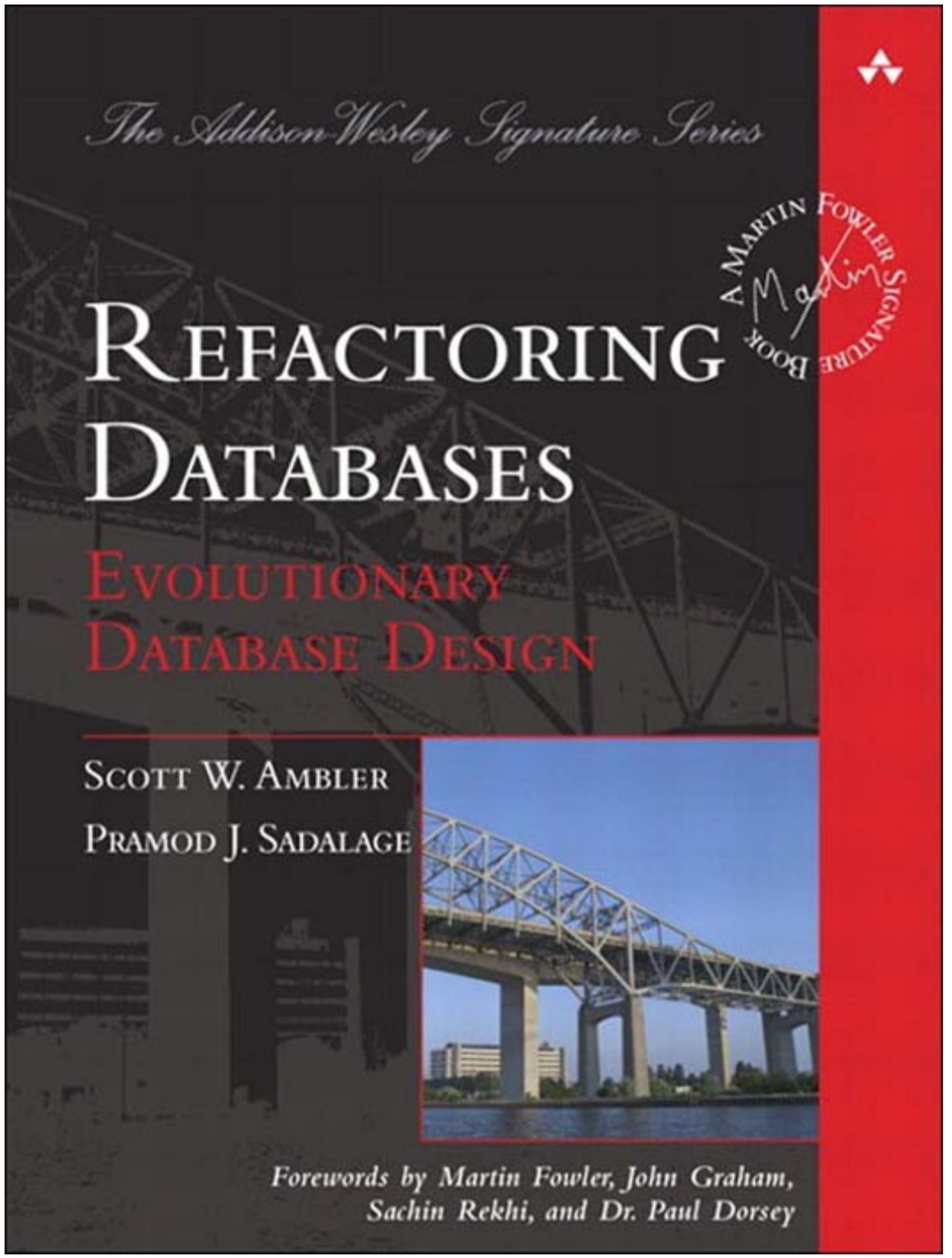
state

long running ……

sagas?

http://www.cs.cornell.edu/andru/cs711/2002fa/reading/sagas.pdf

databases…


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Refactoring Databases
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham,
Sachin Rekhi, and Dr. Paul Dorsey

http://flywaydb.org

environments

love partial failures

do: **build in** switches

Do: apps are **ENV AWARE**

APIs

versioning APIs is hard

different versions, formats, lifecycle

version, document, publish

one place to learn them all

dev env

automate it!

chef, puppet, ansible, docker, vagrant

pick any

Release!

Release!

…the **Kraken**

questions?

@JavaOneConf @ags313 #JavaONE