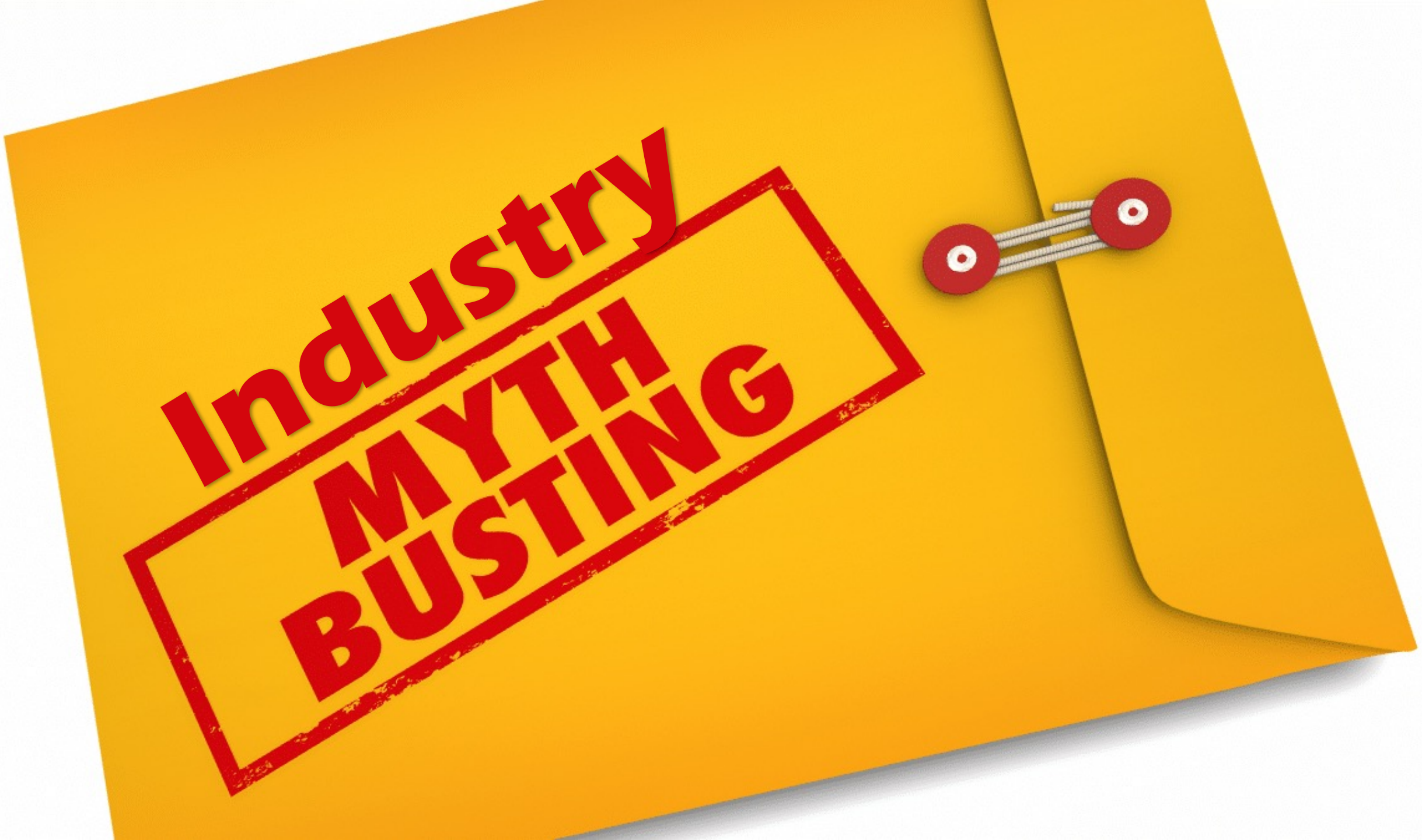
goto;

GOTO
AMSTERDAM 2023

#GOTOams


Industry
MYTH
BUSTING

goto; conference


MYTHBUSTERS

1999

PARTY LIKE IT'S 1999


WIKI


melissa
virus


UNIX
PRINTING
SYSTEM


TLS


SVG


Yeah, well you know,
that's just like your opinion, man.

OLD MAN YELLS AT CLOUD


The picture can't be displayed.

icro-services

Monoliths

2013

Touted Benefits of Microservices

- Speed of development / getting code to production
- Scalability of development
- Choice of technology
- Different lifecycles
- Scalability of individual deployments

Microservices: Our Reasons

For Development:

- Only run what's relevant locally
  - No need to run whole platform
- Only *know* what's relevant
  - Ignore parts you don't know / care about
- Service per client nice granularity
  - Security, focused API, compatibility over time

Microservices: Our Reasons

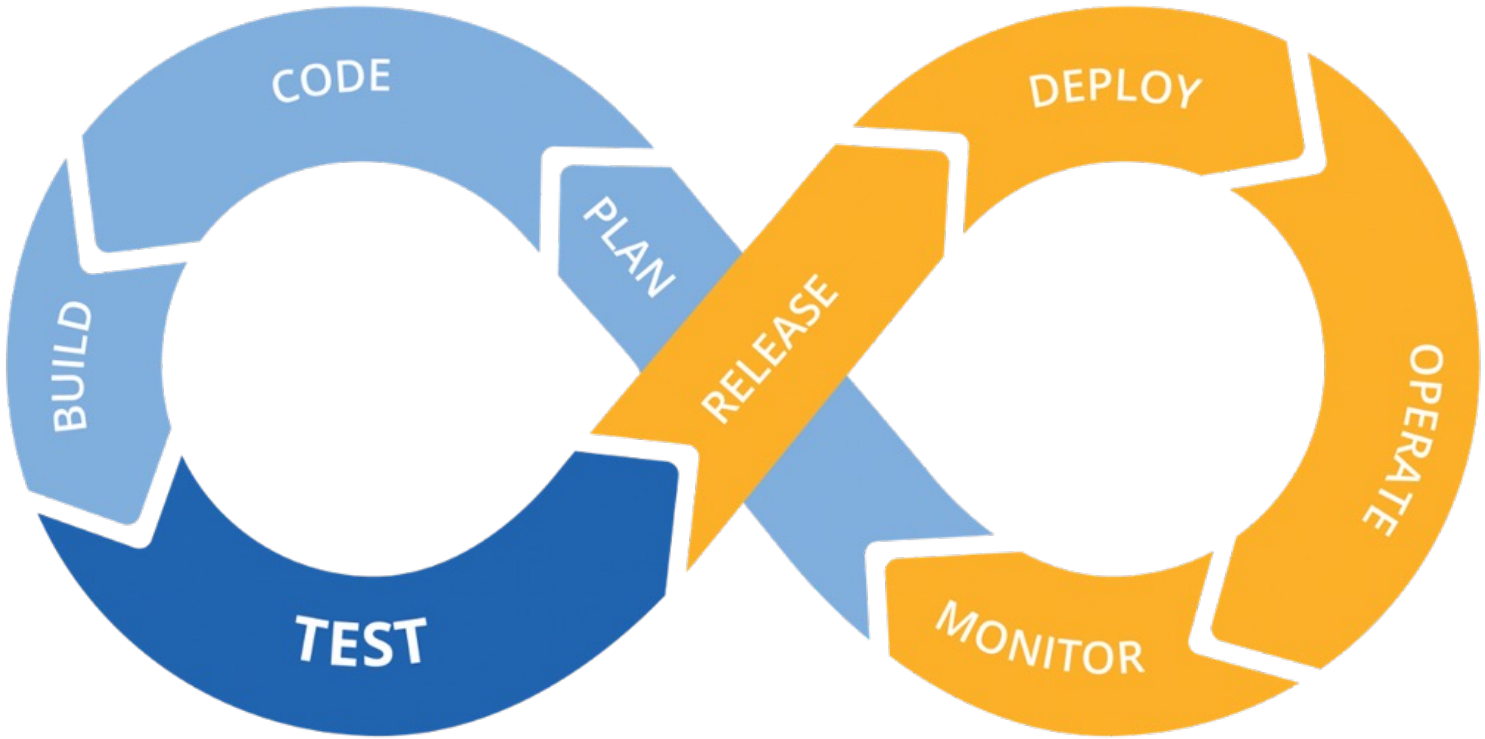
At Runtime:

- Per-service scalability
  - E.g. for Staatsloterij draw or big soccer match
- Limit blast radius when service fails
  - Very important for availability
- Easier to update ad-hoc
  - Just push image &
    update Kubernetes deployment


WHAT ARE
YOUR
GOALS?

Do your constraints still apply?

You must be this tall to use microservices



kubernetes


PROBLEM
SOLVED!

DHH
@dhh

If you're unable to create a majestic monolith with basic programming tools like encapsulation and namespaces, you don't have what it takes to improve upon the situation with a distributed swarm of microservices. Your spaghetti code will just be on five different plates.

9:10 AM · Jun 21, 2023 · **183.2K** Views

394 Retweets **52** Quotes **2,297** Likes **191** Bookmarks

The picture can't be displayed.

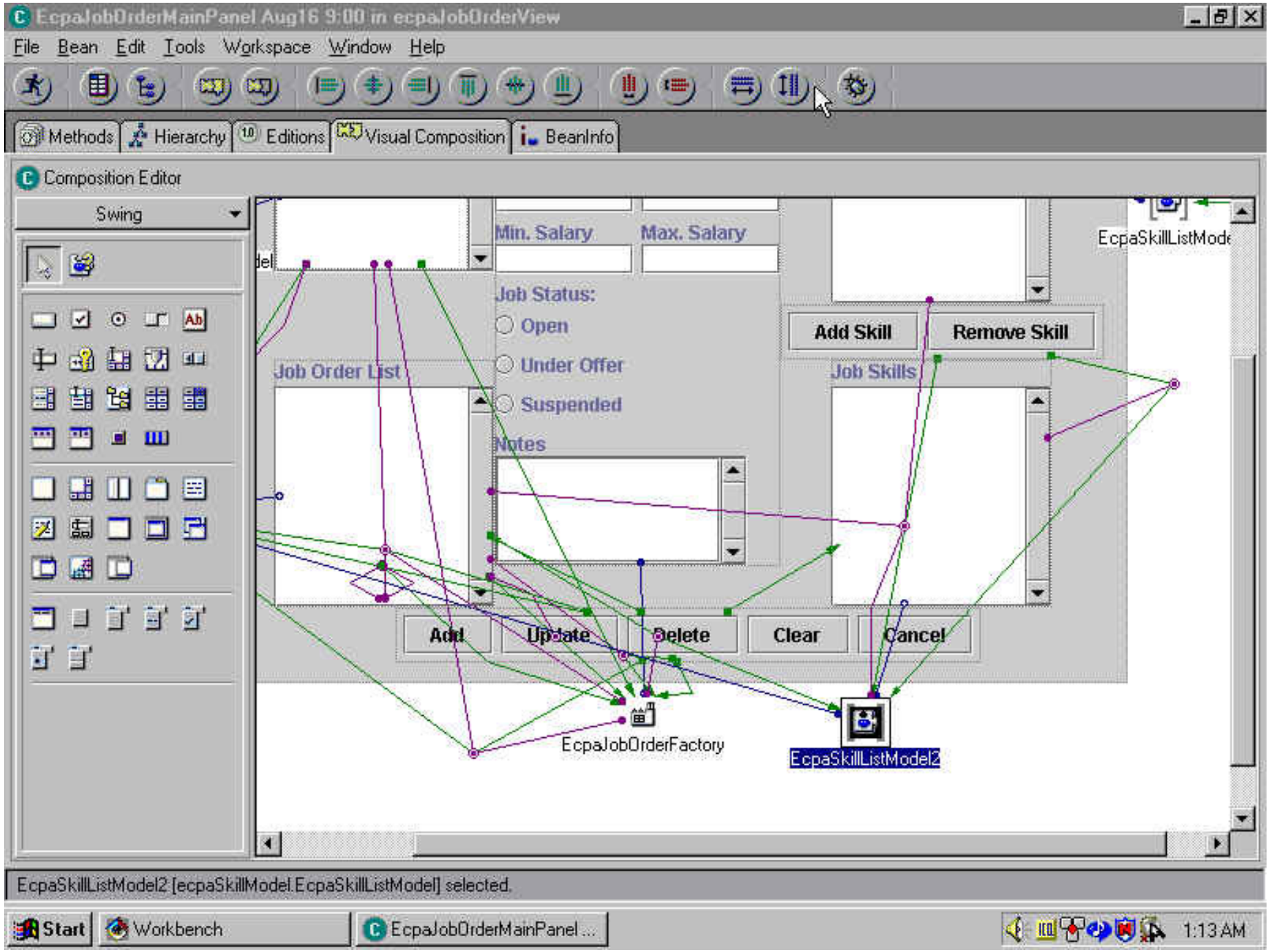
Feature Branches

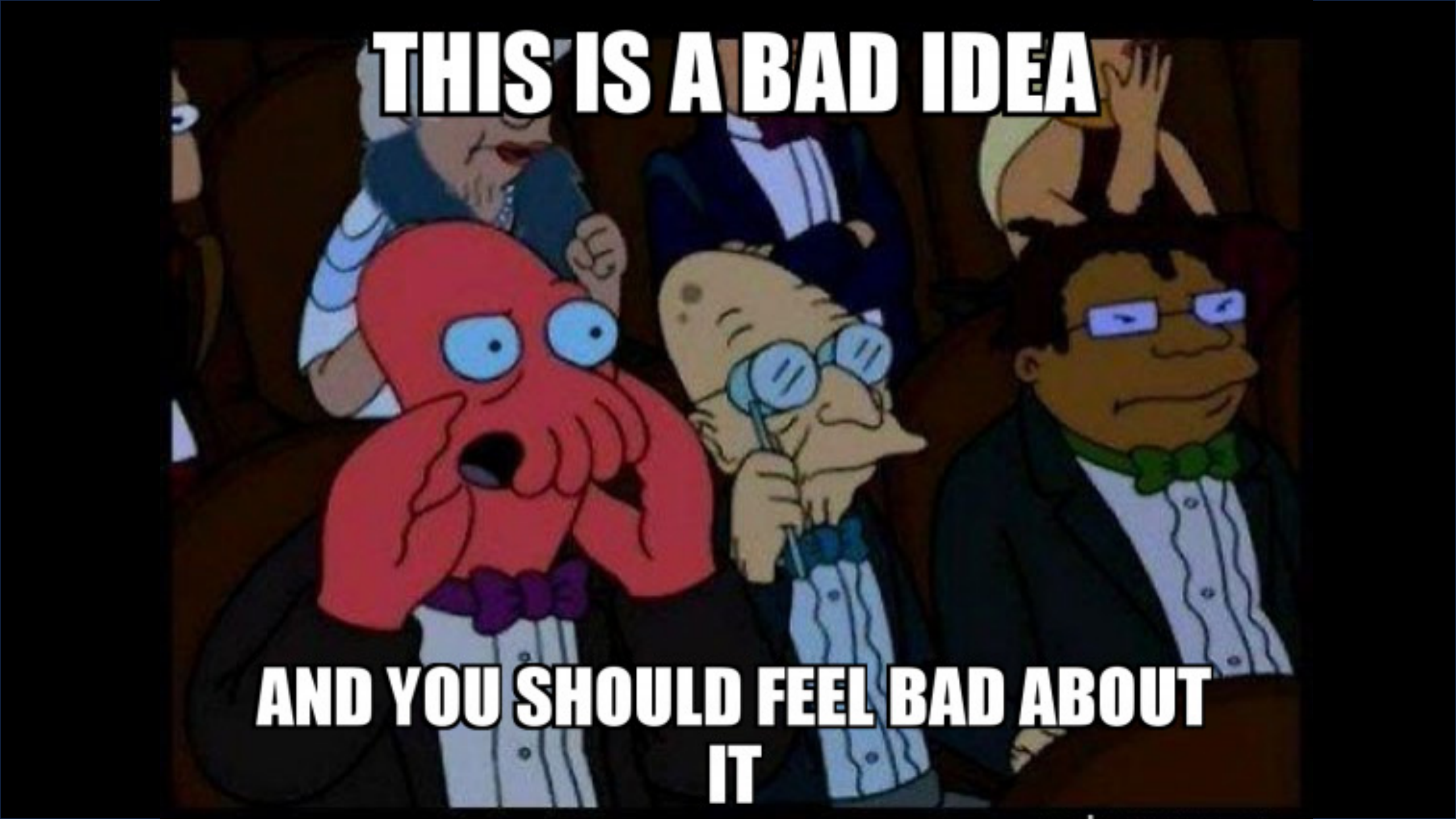
Trunk-based Development

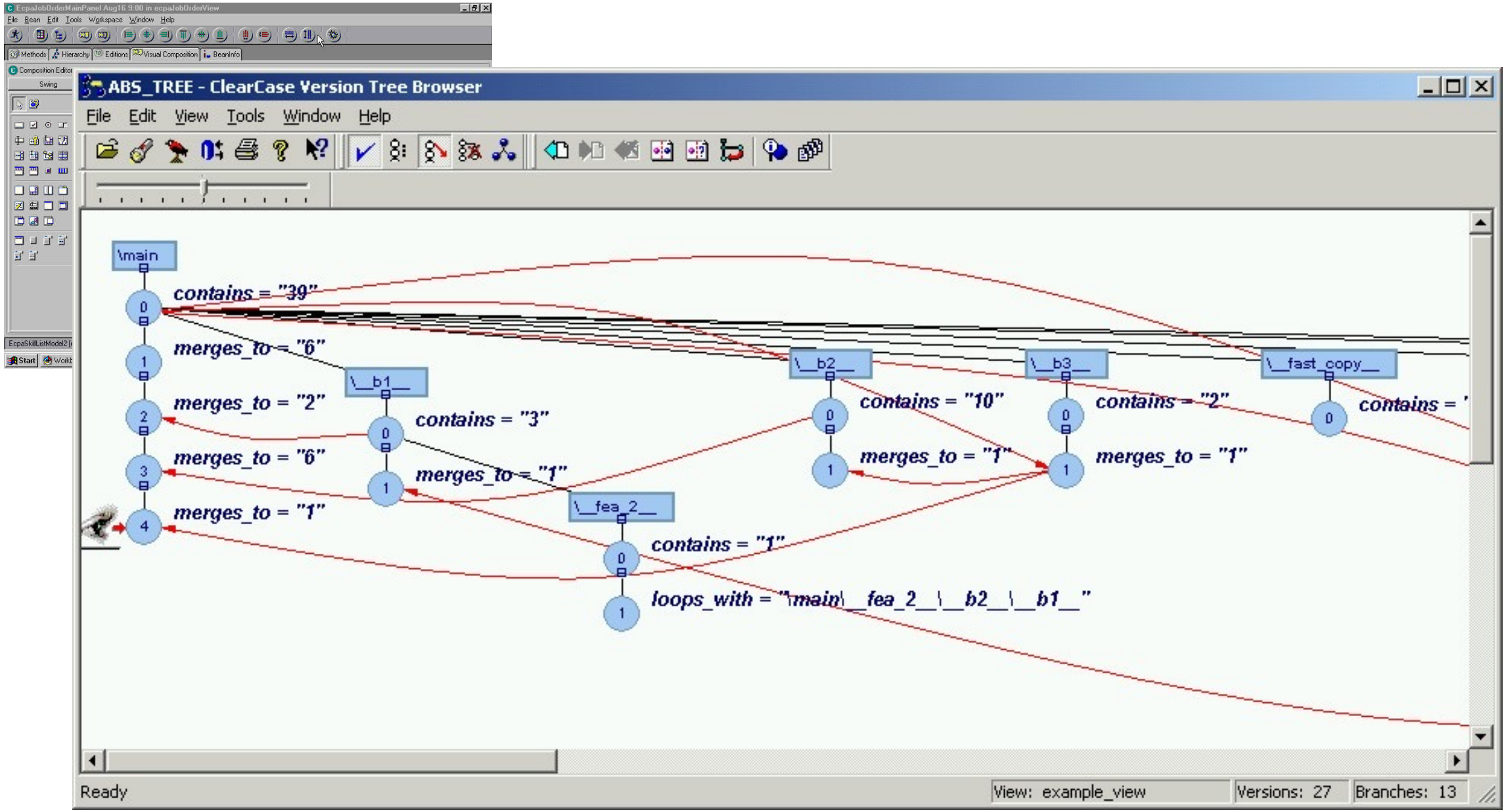
Version Control

Source Control

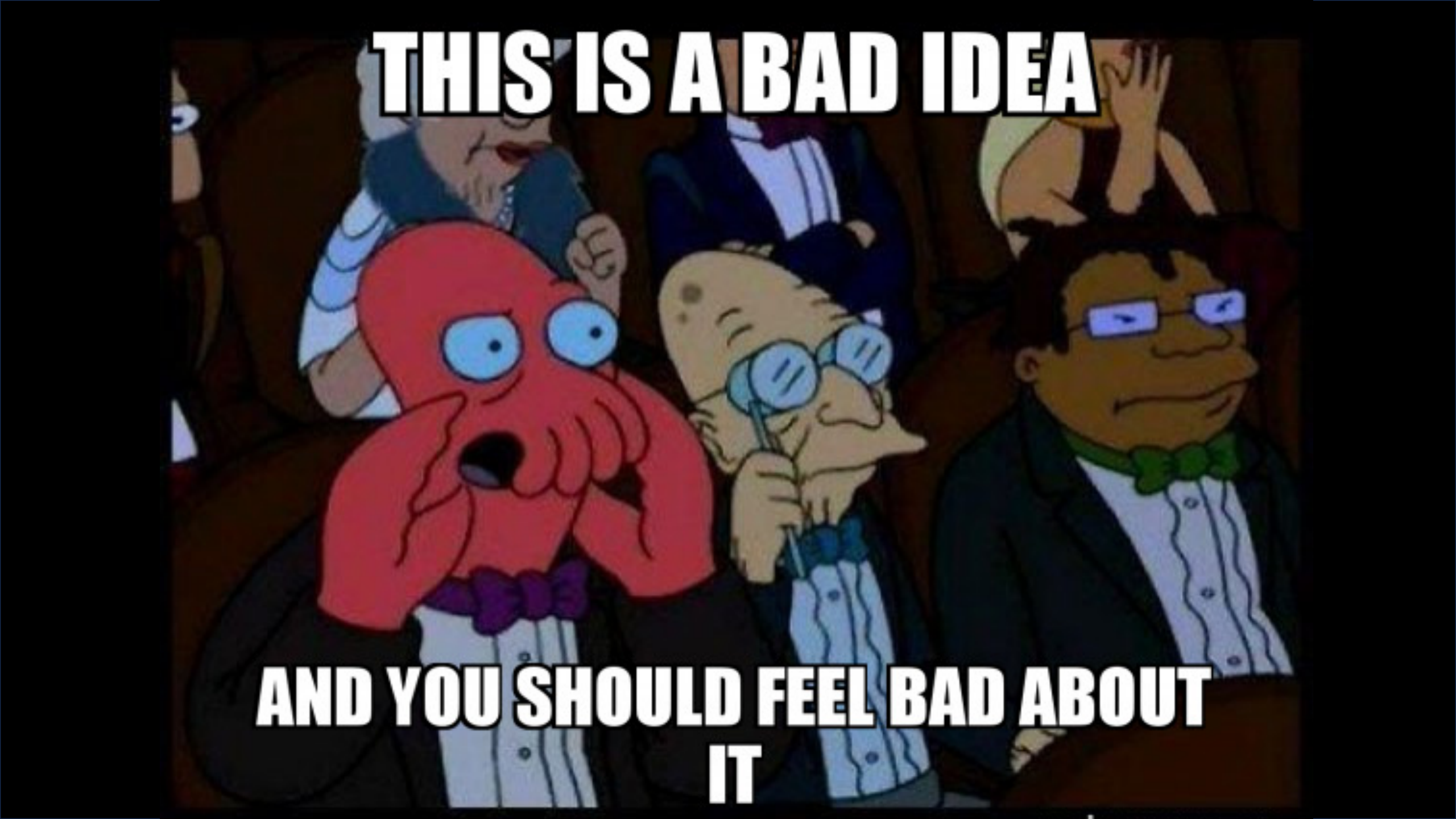
Management


EcpaJobOrderMainPanel Aug16 9:00 in ecpaJobOrderView
File Bean Edit Tools Workspace Window Help
Methods Hierarchy Editions Visual Composition BeanInfo
Composition Editor
Swing
Min. Salary
Max. Salary
Job Status:
Open
Under Offer
Suspended
Notes
Job Order List
Job Skills
Add Skill
Remove Skill
Add
Update
Delete
Clear
Cancel
EcpaSkillListMode
EcpaJobOrderFactory
EcpaSkillListModel2
EcpaSkillListModel2 [ecpaSkillModel.EcpaSkillListModel] selected.
Start
Workbench
EcpaJobOrderMainPanel ...
1:13 AM


THIS IS A BAD IDEA
AND YOU SHOULD FEEL BAD ABOUT IT


ABS_TREE - ClearCase Version Tree Browser
File Edit View Tools Window Help
\main
contains = "39"
merges_to = "6"
merges_to = "2"
merges_to = "6"
merges_to = "1"
__b1__
contains = "3"
merges_to = "1"
__fea_2__
contains = "1"
loops_with = "\main__fea_2____b2____b1__"
__b2__
contains = "10"
merges_to = "1"
__b3__
contains = "2"
merges_to = "1"
__fast_copy__
Ready
View: example_view
Versions: 27
Branches: 13


STAHLHART
BURG
WÄCHTER
Profi


THIS IS A BAD IDEA
AND YOU SHOULD FEEL BAD ABOUT IT

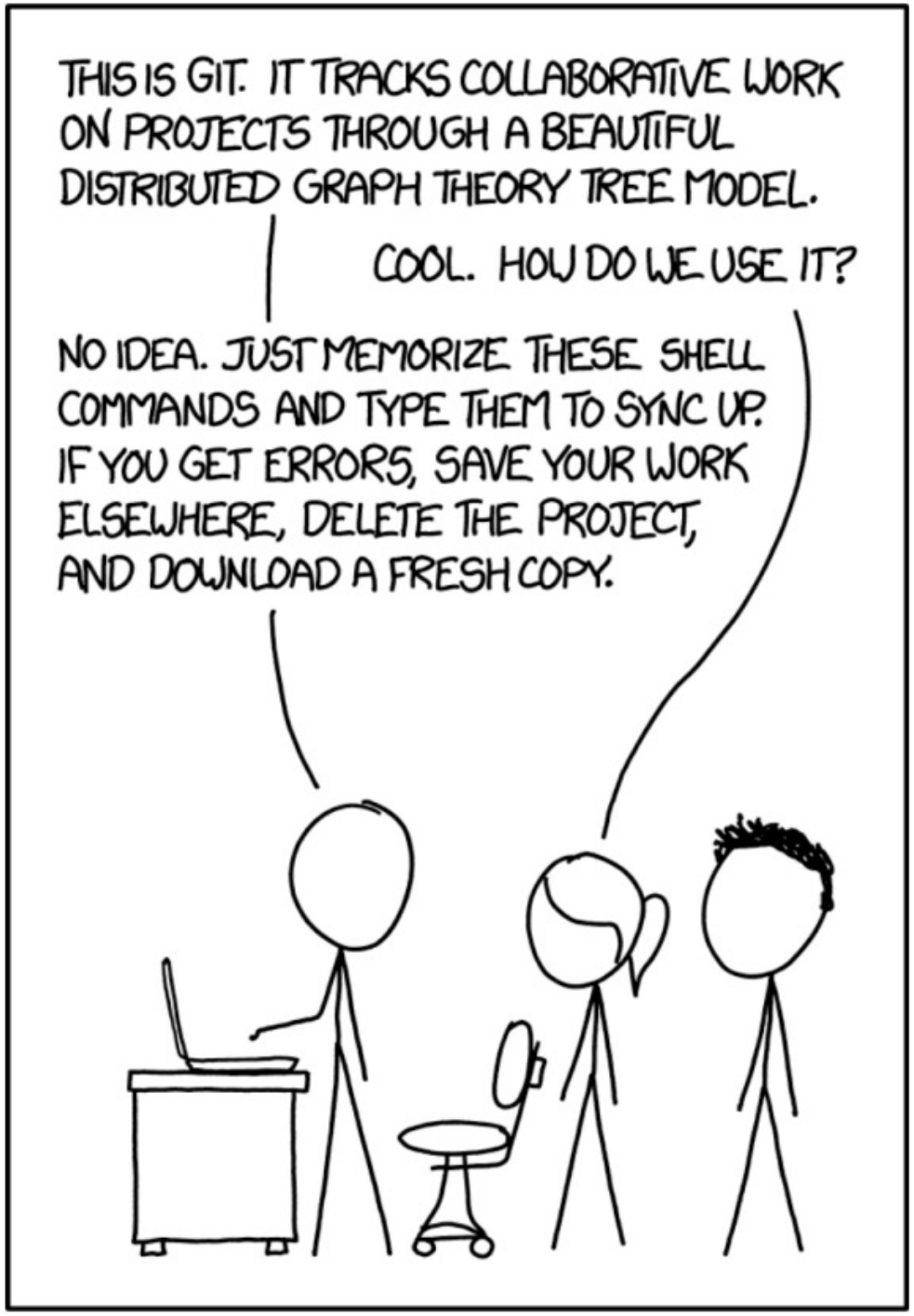

EcpaJobOrderMainPanel Aug16 9:00 in ecpaJobOrderView
File Bean Edit Tools Workspace Window Help


ABS_TREE - ClearCase Version Tree Browser
File Edit View Tools Window Help

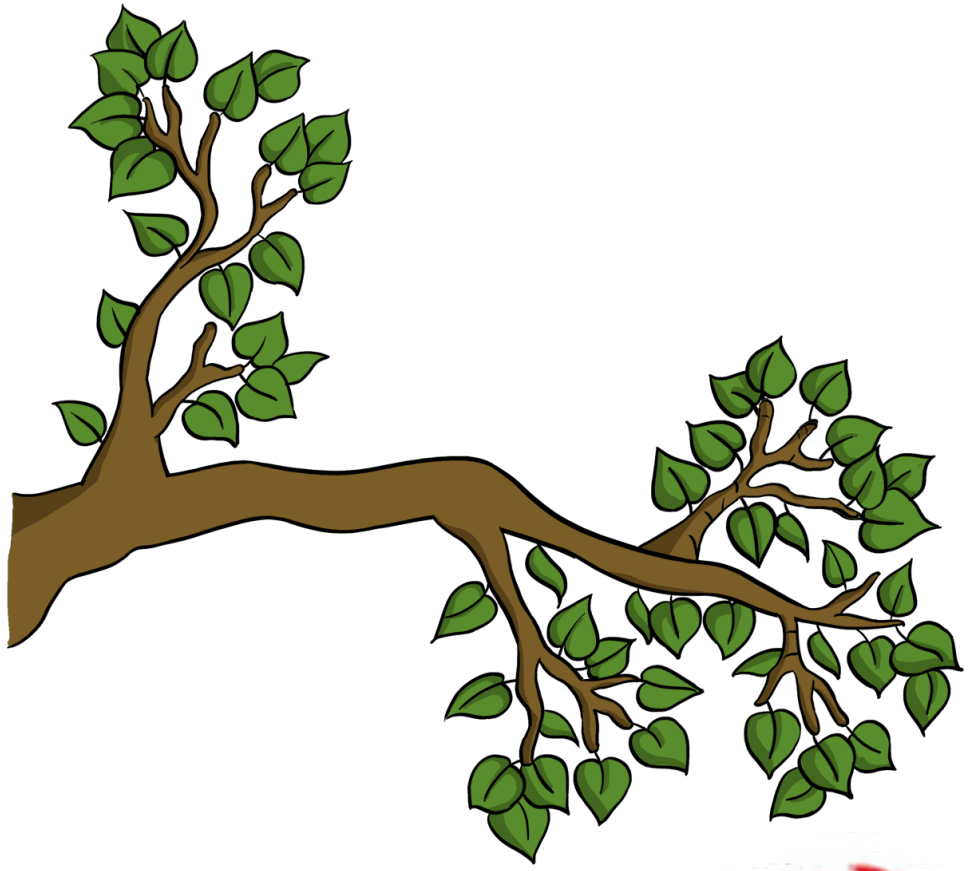

git

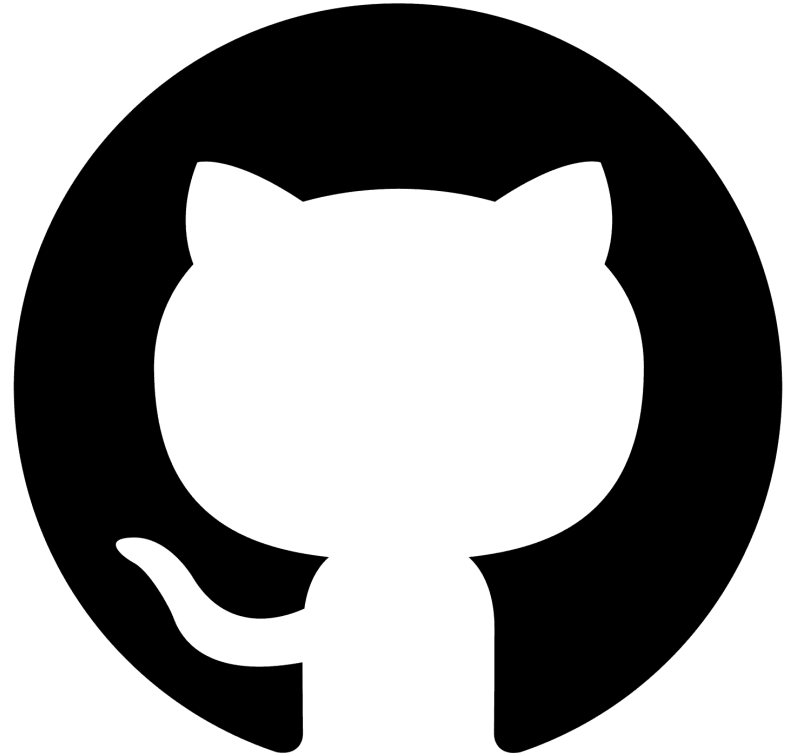
"Git gets easier once you get the basic idea that branches are homeomorphic endo-functors mapping submani-folds for a Hilbert space."

-- internet joke

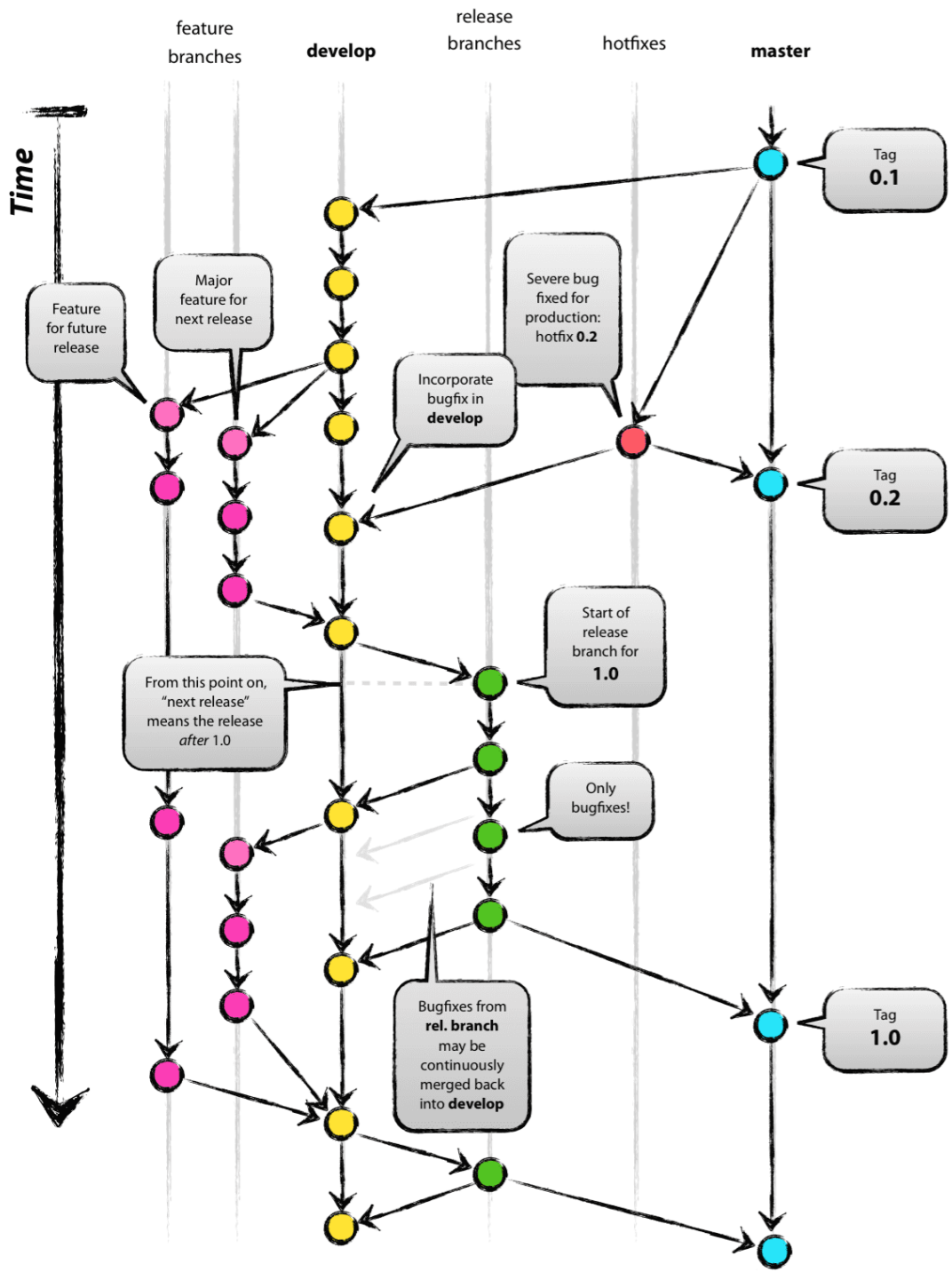


https://xkcd.com/1597/





GitHub


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Trunk Based Development

“If you’re not doing trunk-based development, you’re not doing continuous integration”

-- zealots, typically


¯_(ツ)_/¯

“If you’re not using
hypermedia,
your API is not RESTful”

-- people who are right
but not helpful, typically

Ship / Show / Ask

A modern branching strategy

Ship/Show/Ask is a branching strategy that combines the features of Pull Requests with the ability to keep shipping changes. Changes are categorized as either Ship (merge into mainline without review), Show (open a pull request for review, but merge into mainline immediately), or Ask (open a pull request for discussion before merging).

08 September 2021

Rouan Wilsenach

CONTENTS

What Affects Branching Strategies?

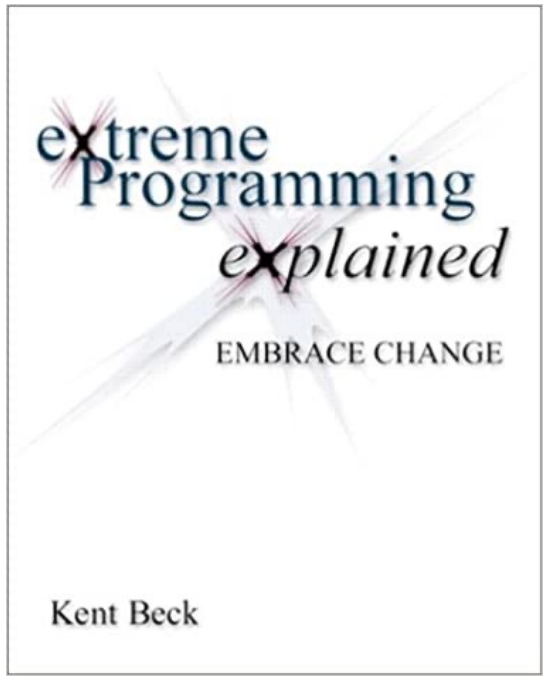
- Feature release frequency
- Latest & greatest or stabilize via release branch
- DB Schema changes and other migrations
- Ability to use feature flags
- Governance / auditing requirements


WHAT ARE
YOUR
GOALS?

Do your constraints still apply?

Developer Testing

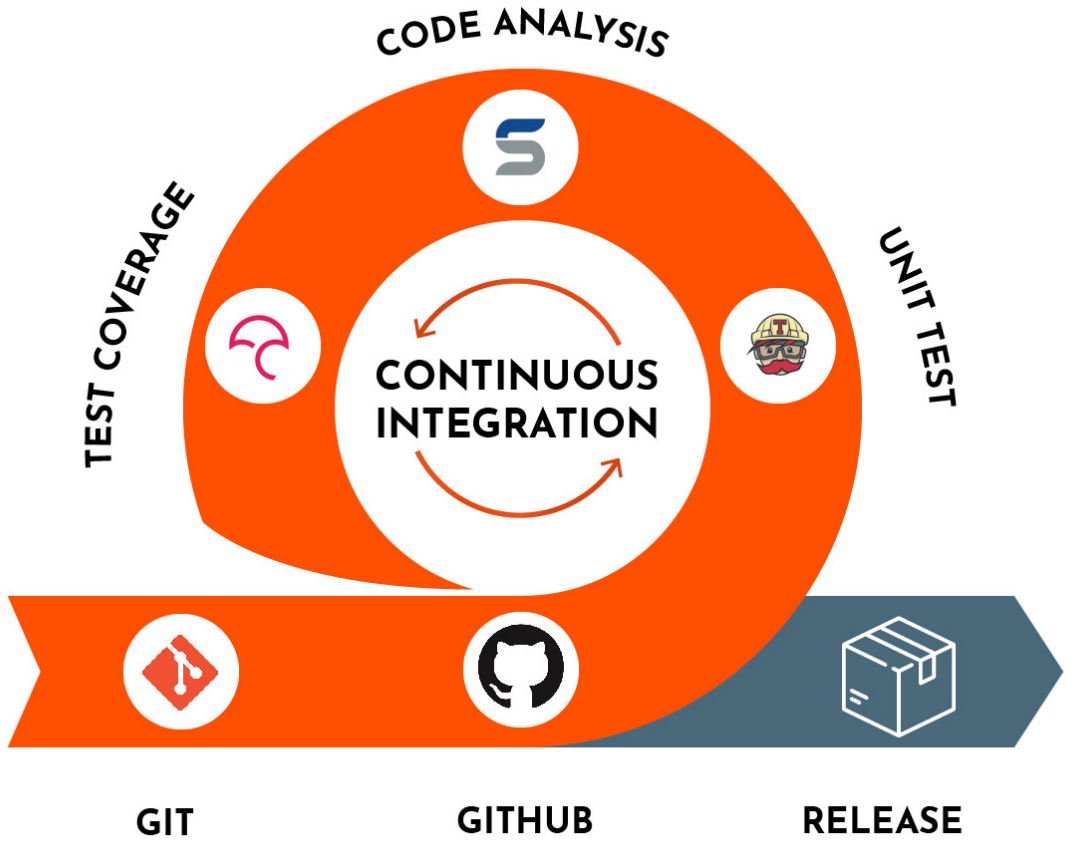

JUnit


extreme
Programming
explained
EMBRACE CHANGE
Kent Beck

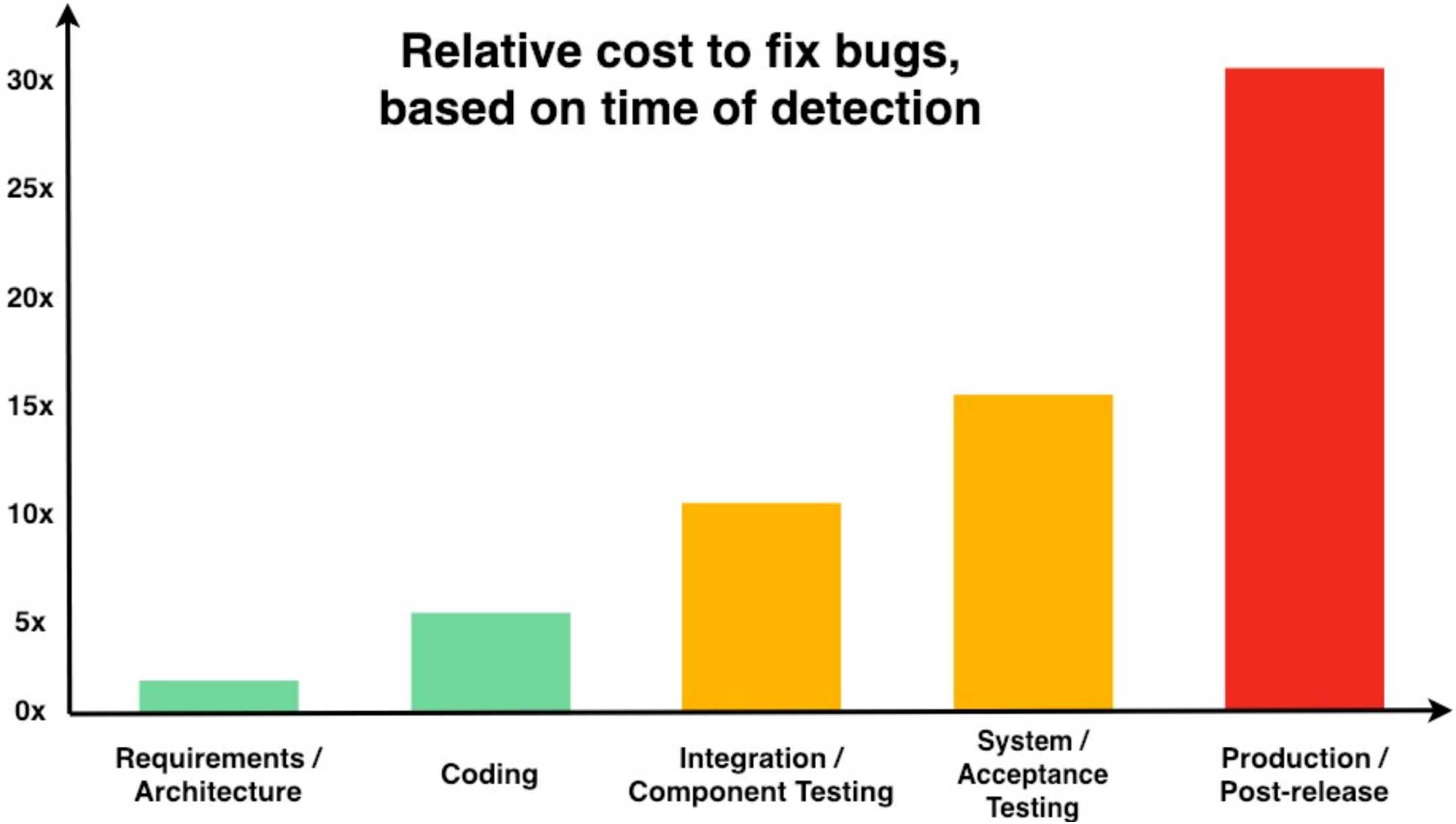

Run Test
Write Code
Refactor and Repeat
Test Driven
Development

%

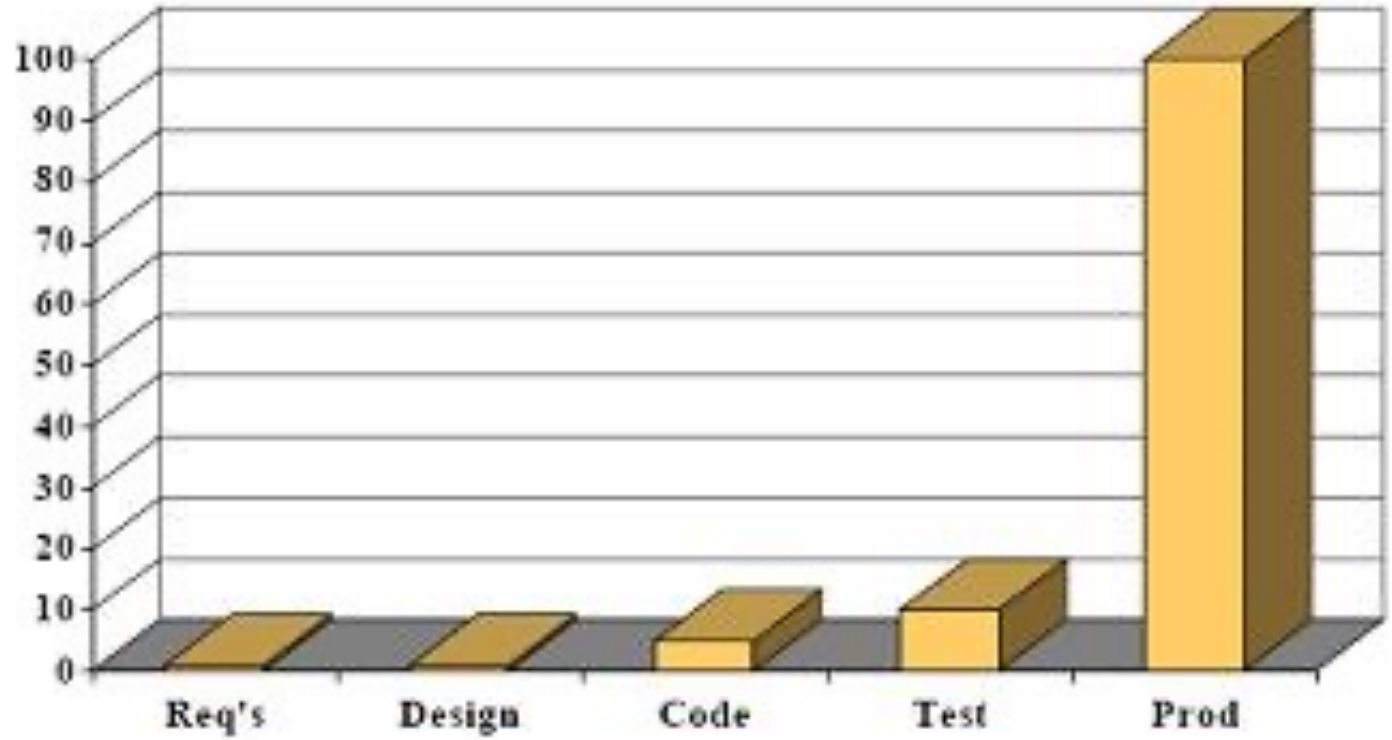
Coverage

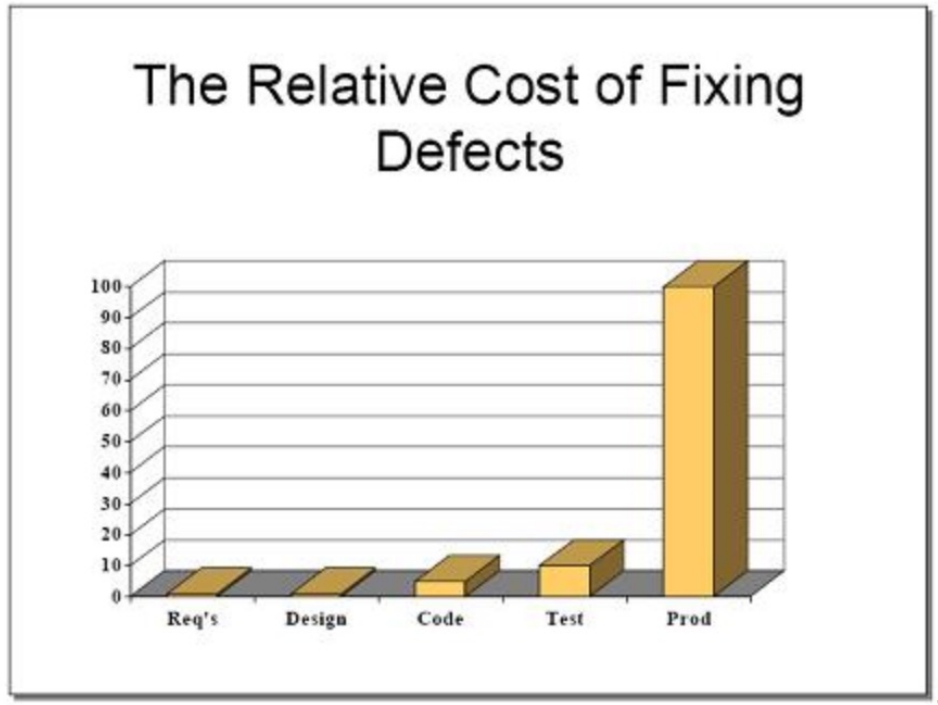

CODE ANALYSIS
TEST COVERAGE
UNIT TEST
CONTINUOUS
INTEGRATION
GIT
GITHUB
RELEASE

$$\text{Return on Investment (ROI)} = \frac{\text{Net Return}}{\text{Cost of Investment}}$$


Relative cost to fix bugs,
based on time of detection
30x
25x
20x
15x
10x
5x
0x
Requirements / Architecture
Coding
Integration / Component Testing
System / Acceptance Testing
Production / Post-release

The Relative Cost of Fixing Defects





Source:

http://www.riceconsulting.com/public_pdf/STBC-WM.pdf

If this doesn't convince you of the value of unit tests, I don't know what will. The cost to fix bugs in production could be dramatically higher than the cost to fix them in development, which is why having a suite of unit tests that you can run when you make changes is invaluable because it will prevent you from getting into these costly scenarios where you have to fix nasty bugs in production.

COMMENTS

REPLY

BLINDMAN

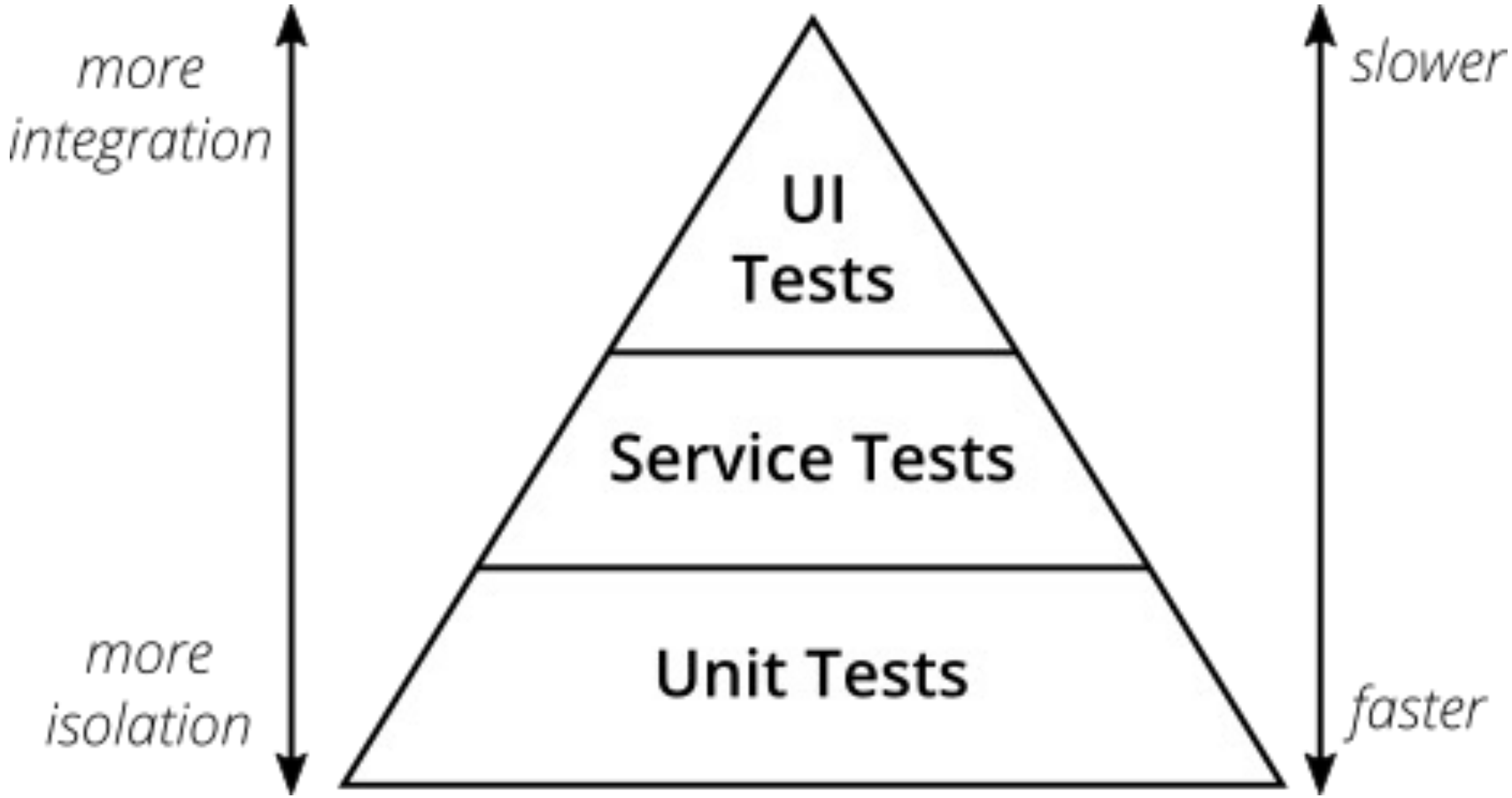
November 21, 2008 at 10:45 am

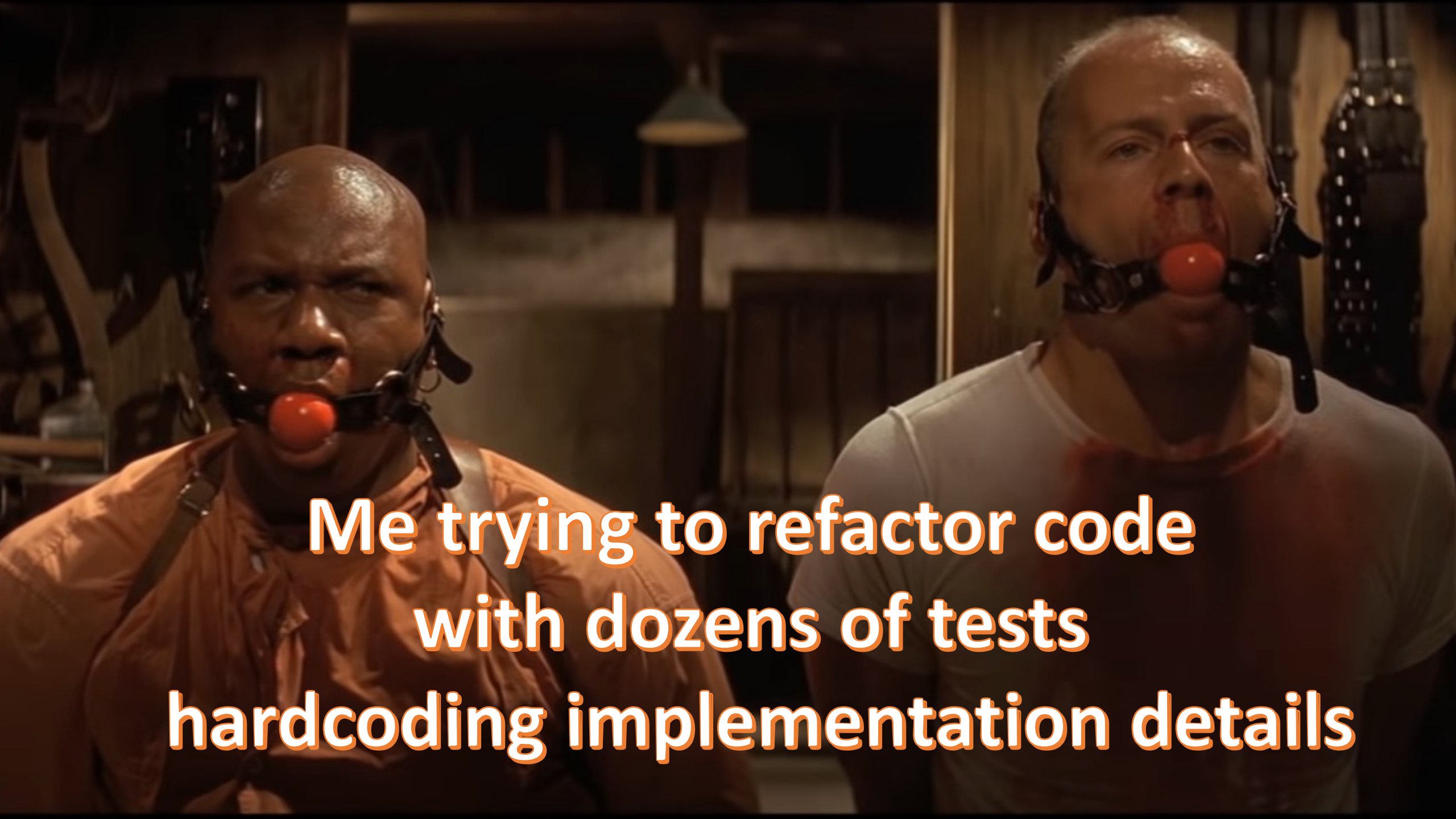
"If this doesn't convince you of the value of unit tests, I don't know what will."

Absolutely. I am a sucker for nice 3D bar graphs, and basically believe anything they tell me.

Also….brown eyes. I never could resist them.

Test Pyramid (Mike Cohn, 2009)




Me trying to refactor code
with dozens of tests
hardcoding implementation details

Gavin King 🎹🎹🎹 @1ovthafew · Nov 1, 2022

Still feel like some folks have a too-simplistic model of testing, where more tests => better. I prefer to think in terms of false negatives vs. false positives. A false + is where a change to internal impl fails lots of tests. If your tests give lots of those, they're bad tests.

1 4 9

Gavin King 🎹🎹🎹
@1ovthafew

False positives are a tax on refactoring. And the deadweight loss associated with that tax is that developers will be incentivized to do less cleanup work on bad code.

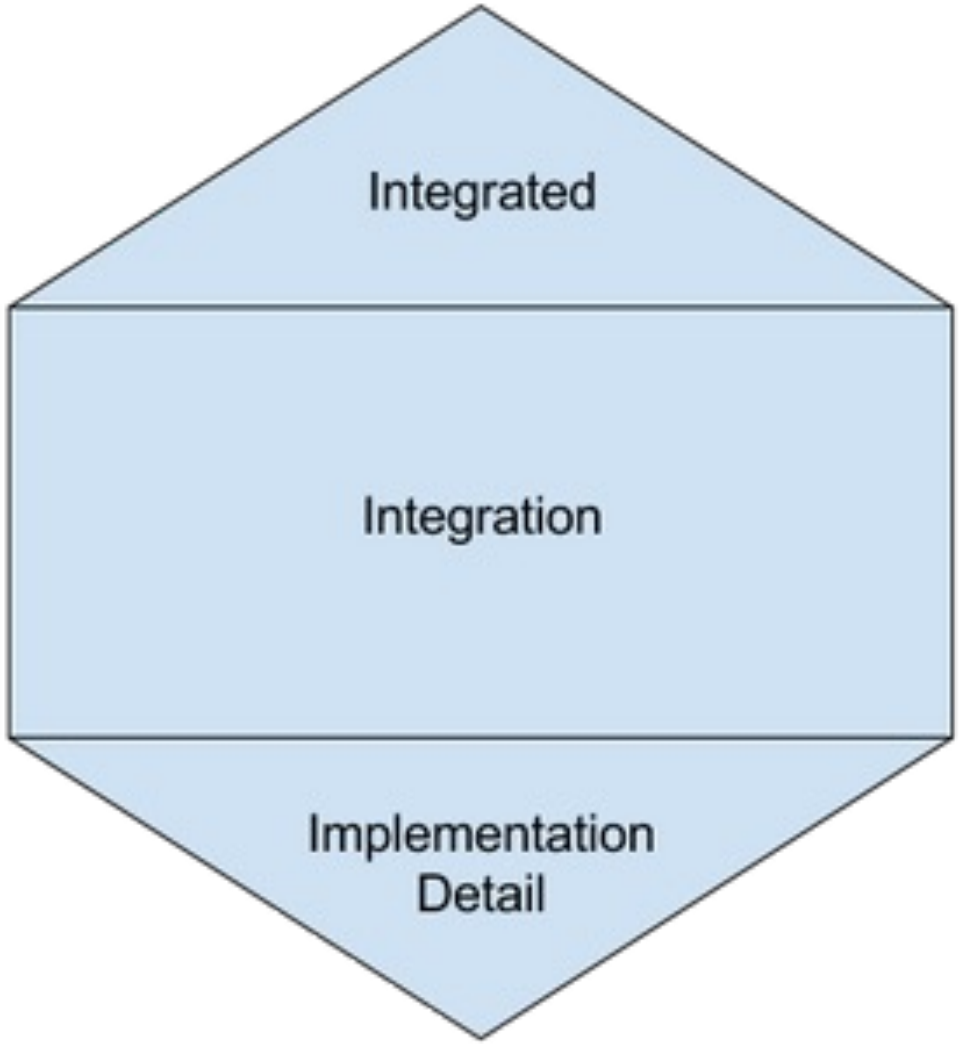
1:09 PM · Nov 1, 2022

How many code reviews have lead to *less* unit tests?

How often do you *delete* tests?


Me in a project where
customer checks the promised
X% code coverage for every sprint

Test Honeycomb (André Schafer, 2018)




Unit test vs. Integration test


WHAT ARE
YOUR
GOALS?

Do your constraints still apply?

Graeme Roche... @graemer... · Nov 18, 2019 •••

Right, perhaps. I have personally seen too many projects give up on writing functional tests because the app takes too long to start. That is a shame IMO as it is the framework dictating to you the kind of tests you can write

Testcontainers

This Also Applies To:

- Interfaces vs concrete classes
- Services vs serverless
- The Right Programming Language
- *My prediction:*
  Artisanal vs AI-generated code
- …

The picture can't be displayed.

So now what?

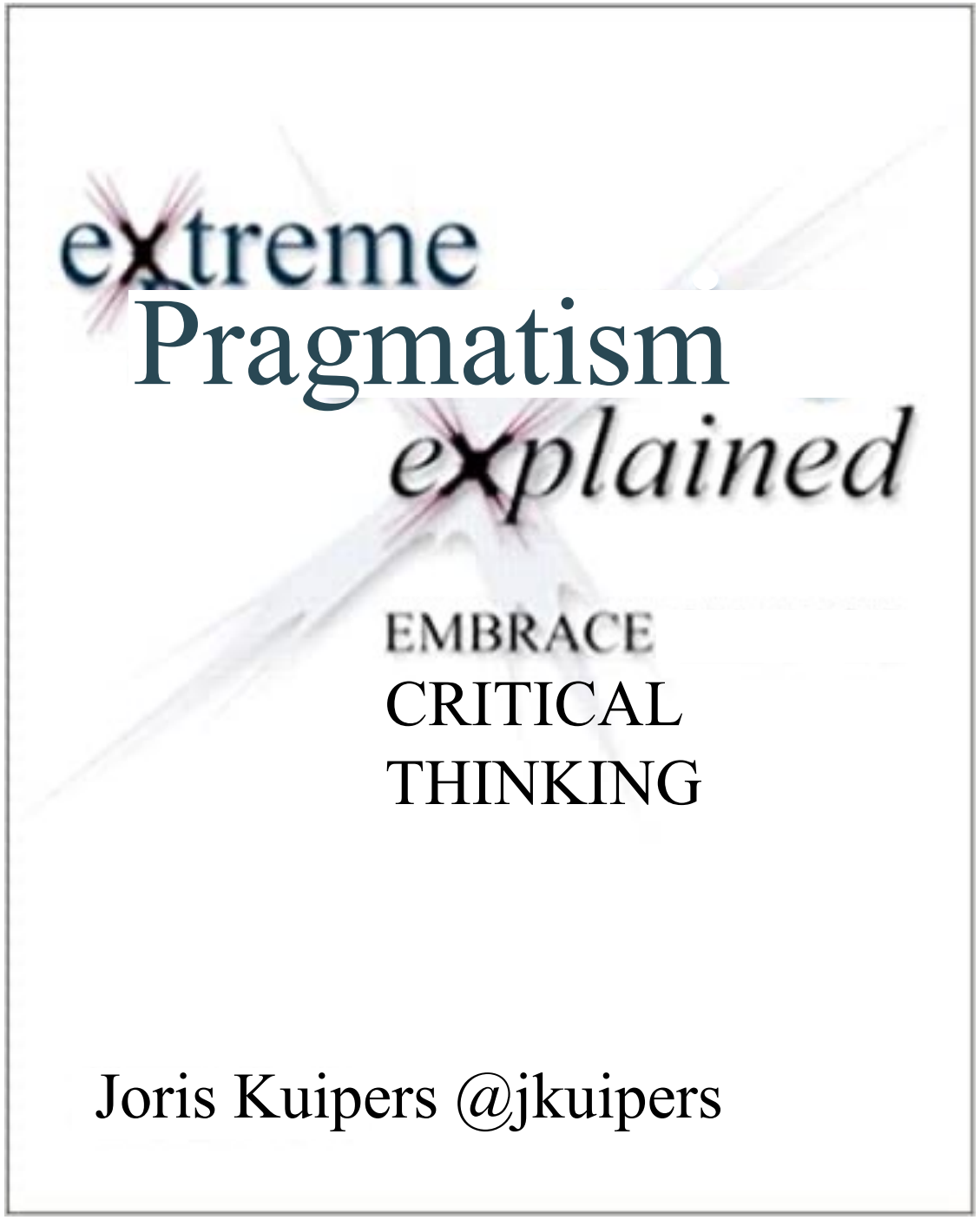

NO, NO. HE'S GOT A POINT.

“You need to
understand their context
and then figure out
how that applies to you”

-- me

Joris Kuipers
@jkuipers

#CloudExpat:
🏙️ you weren't born in the cloud, you're just there to work
😩 you complain about the things that seem more complex than they were back where you came from
🤪 you make fun of all the idiosyncrasies of the natives
💸 you pay less taxes than the locals


extreme
Pragmatism
explained
EMBRACE
CRITICAL
THINKING
Joris Kuipers @jkuipers

GOTO
Guide

★★★★★

Remember to **rate this session**

THANK YOU!

#GOTOams