ThoughtWorks®

IT'S NOT CONTINUOUS DELIVERY

If you can't deploy to production right now

WHO AM I?

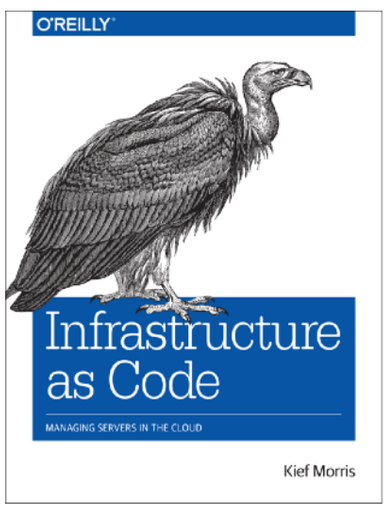
Ken Mugrage

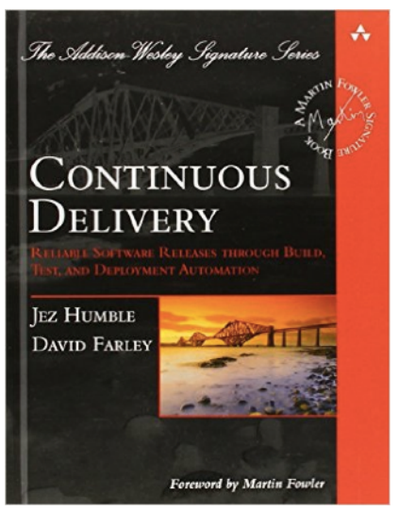
ThoughtWorks Technology Evangelist

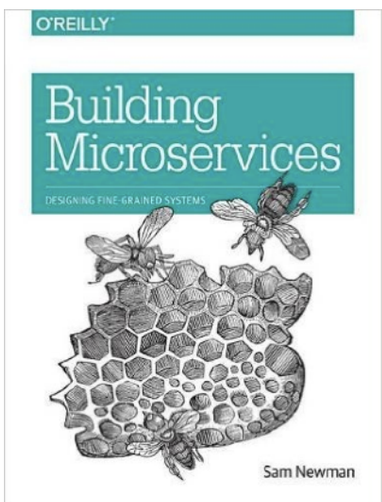
@kmugrage

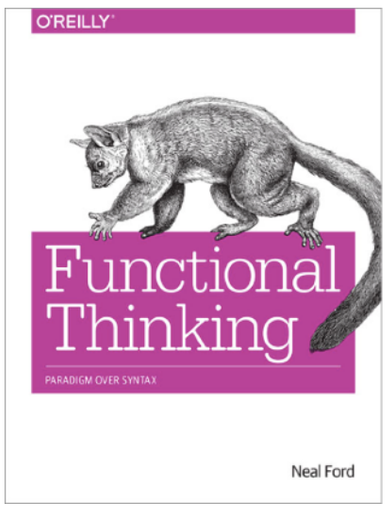
$11,892 **@kmugrage**

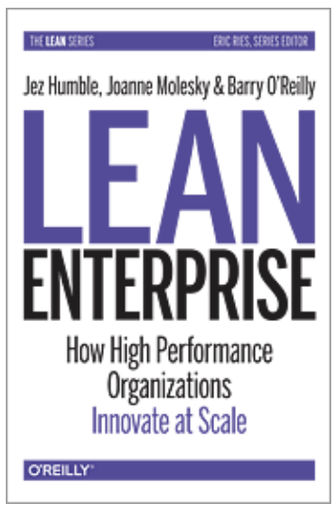
OPINIONATED THOUGHT LEADERSHIP











$23,784

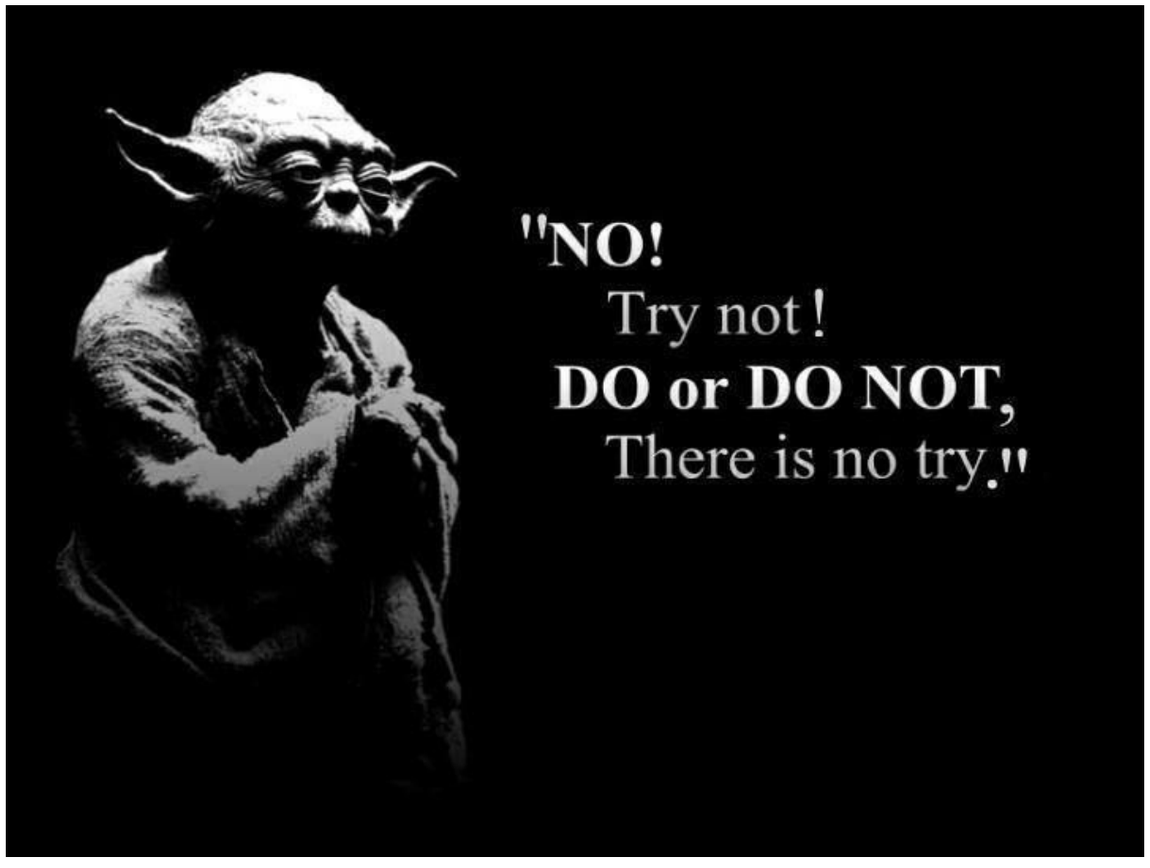
@kmugrage

ThoughtWorks®

WHY THIS TALK

$35,676

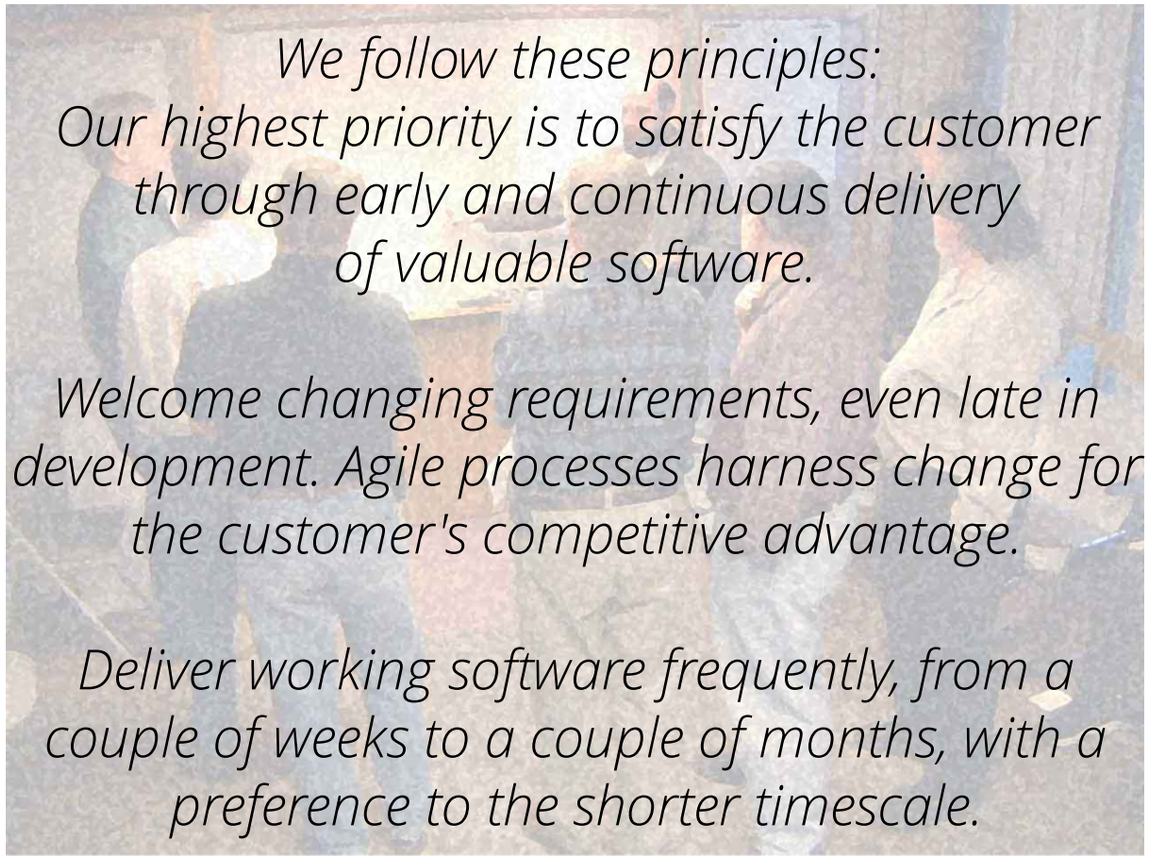
THERE IS NO TRY



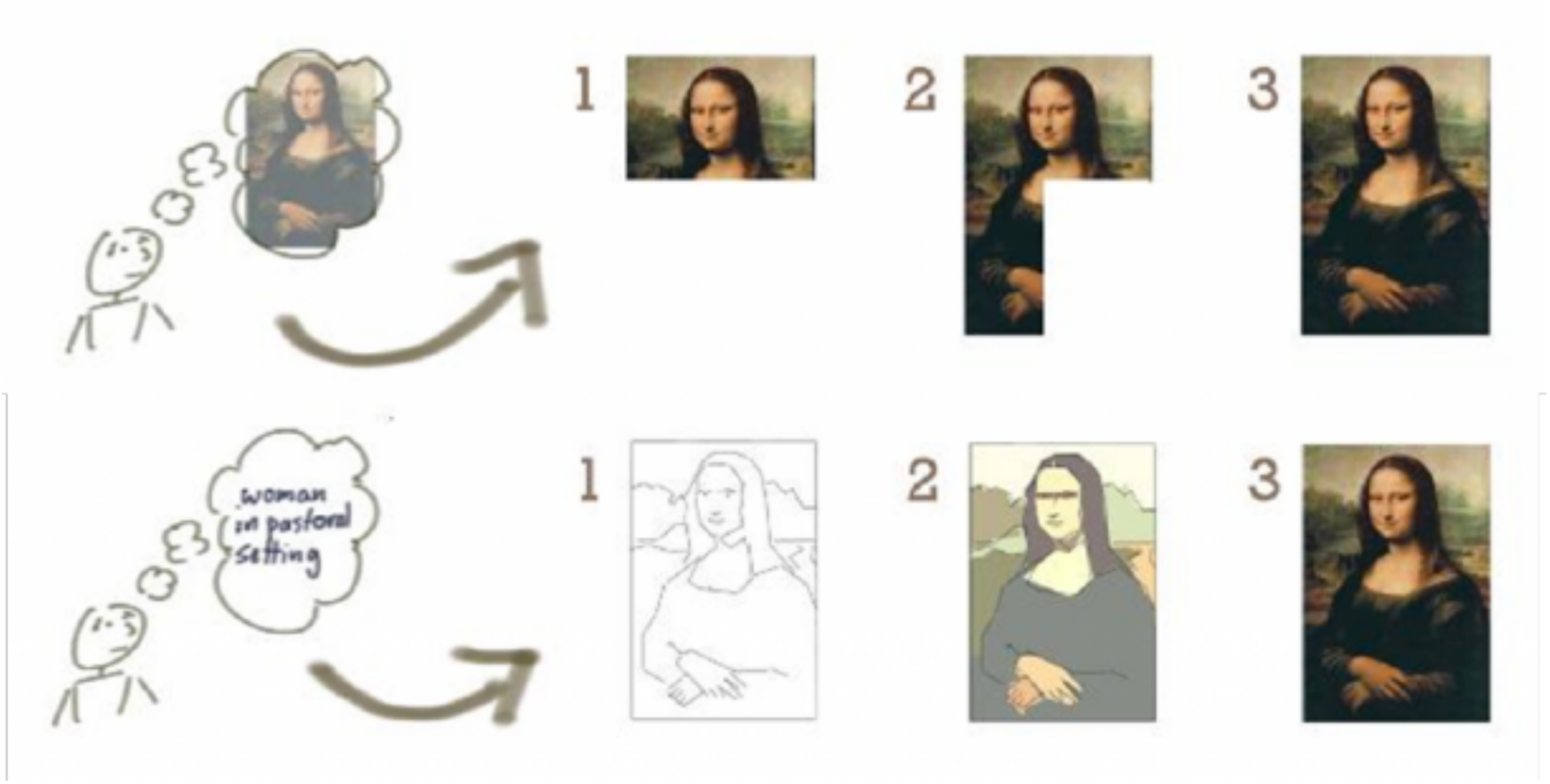
$47,568

@kmugrage

WHY CONTINUOUS DELIVERY?



PARTIALLY “DONE” MIGHT STILL BE USEFUL



$71,352

@kmugrage

RESPOND TO SECURITY ISSUES

At the time of disclosure, some 17% (around half a million) of the Internet's secure web servers certified by trusted authorities were believed to be vulnerable to the attack, allowing theft of the servers' private keys and users' session cookies and passwords.

$83,244

https://en.wikipedia.org/wiki/Heartbleed

@kmugrage

KNIGHT CAPITAL

To be continued....

$95,136

@kmugrage

ThoughtWorks®

CONTINUOUS INTEGRATION

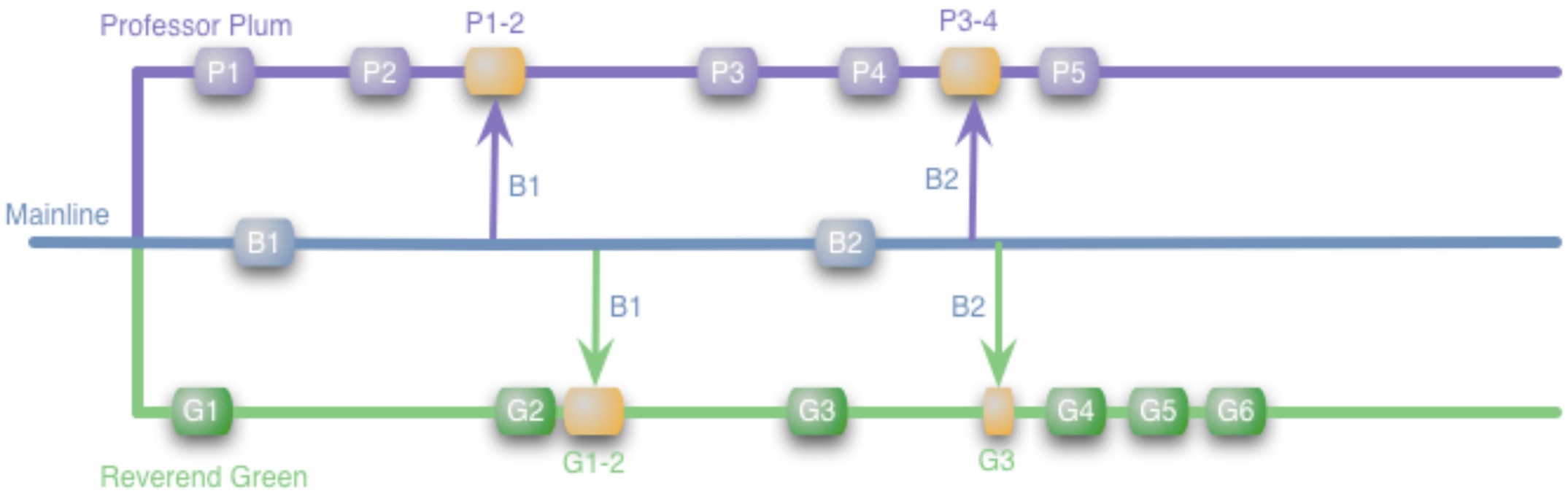
$107,028

The ThoughtWorks tech radar recently recommended putting a hold on the tech team anti-pattern, CI Theatre. CI Theatre describes the illusion of practicing continuous integration (CI) while not really practicing it.

$118,920 *https://www.gocd.org/2017/05/16/its-not-CI-its-CI-theatre/* **@kmugrage**

In our study only 10% of participants acknowledged that having a CI server was not the same as practicing CI.

FEATURE BRANCHING

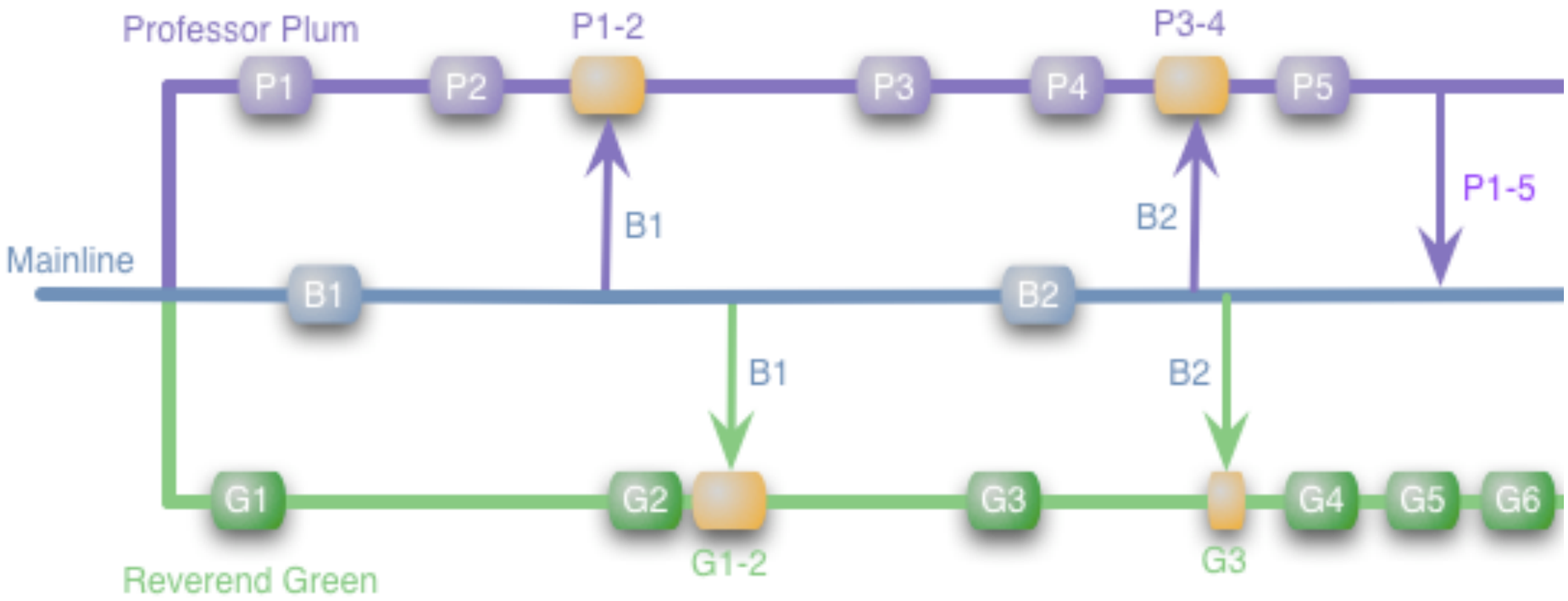


$142,704

https://martinfowler.com/bliki/FeatureBranch.html

@kmugrage

FEATURE BRANCHING

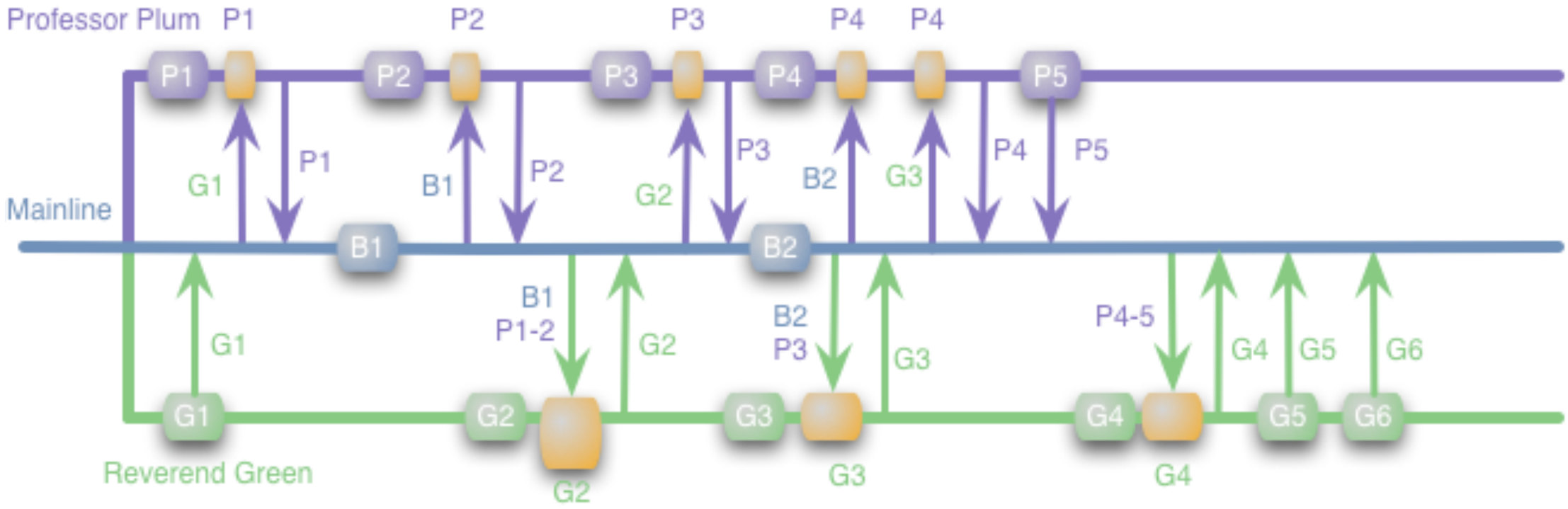


$154,596

https://martinfowler.com/bliki/FeatureBranch.html

@kmugrage

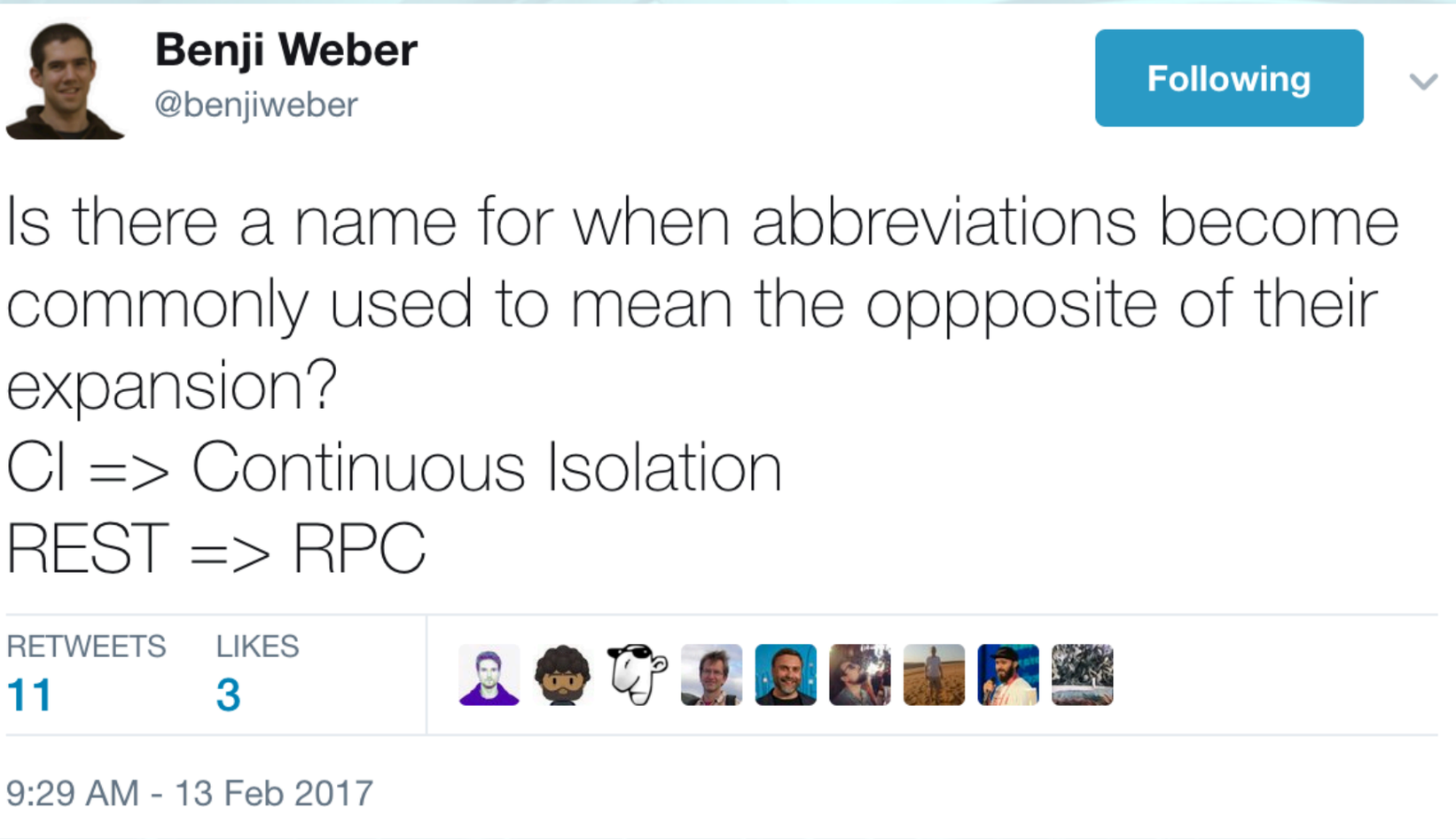
FEATURE BRANCHING



$166,488

https://martinfowler.com/bliki/FeatureBranch.html

@kmugrage



$178,380

@kmugrage

RECOMMENDED CI PRACTICES

Commit more often

Trunk based development

Automate

ThoughtWorks®

PIPELINES YOU SHOULD BE INCLUDING

$202,164

SECURITY TESTING

Test before you commit

Have you included private keys? Authentication tokens?

Static Application Security Testing (SAST)

According to one Sonatype study “of the 106 component ‘parts’ used in a typical application, on average 24 have known cyber vulnerabilities, which are rated either critical or severe."

Dynamic Application Security Testing (DAST)

Tools that run against your code are a good start, but they aren’t accessing the application like a user.

Pre-commit checks: https://thoughtworks.github.io/talisman/

@kmugrage

$214,056

PERFORMANCE TESTING

Load testing

Load testing is the simplest form of performance testing. A load test is usually conducted to understand the behavior of the system under a specific expected load.

Stress testing

Stress testing is normally used to understand the upper limits of capacity within the system.

Soak testing

Soak testing, also known as endurance testing, is usually done to determine if the system can sustain the continuous expected load.

Spike testing

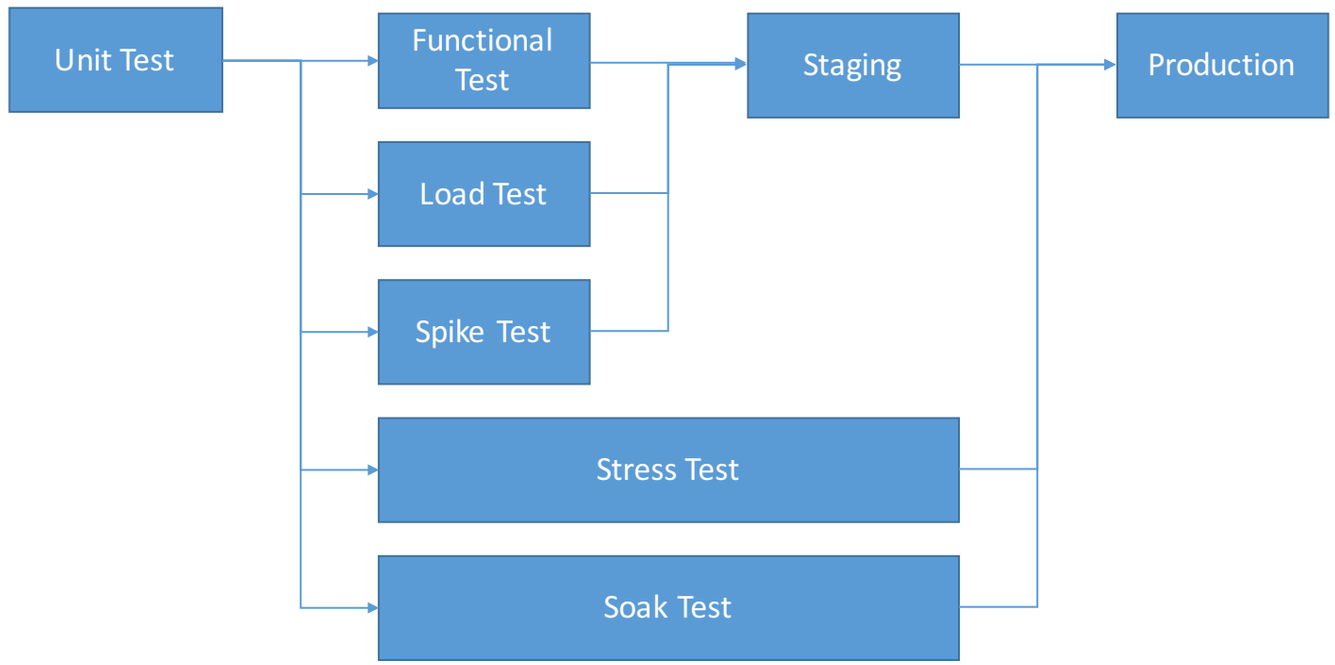
Spike testing is done by suddenly increasing the load generated by a very large number of users, and observing the behavior of the system.

@kmugrage

$225,948

https://en.wikipedia.org/wiki/Software_performance_testing

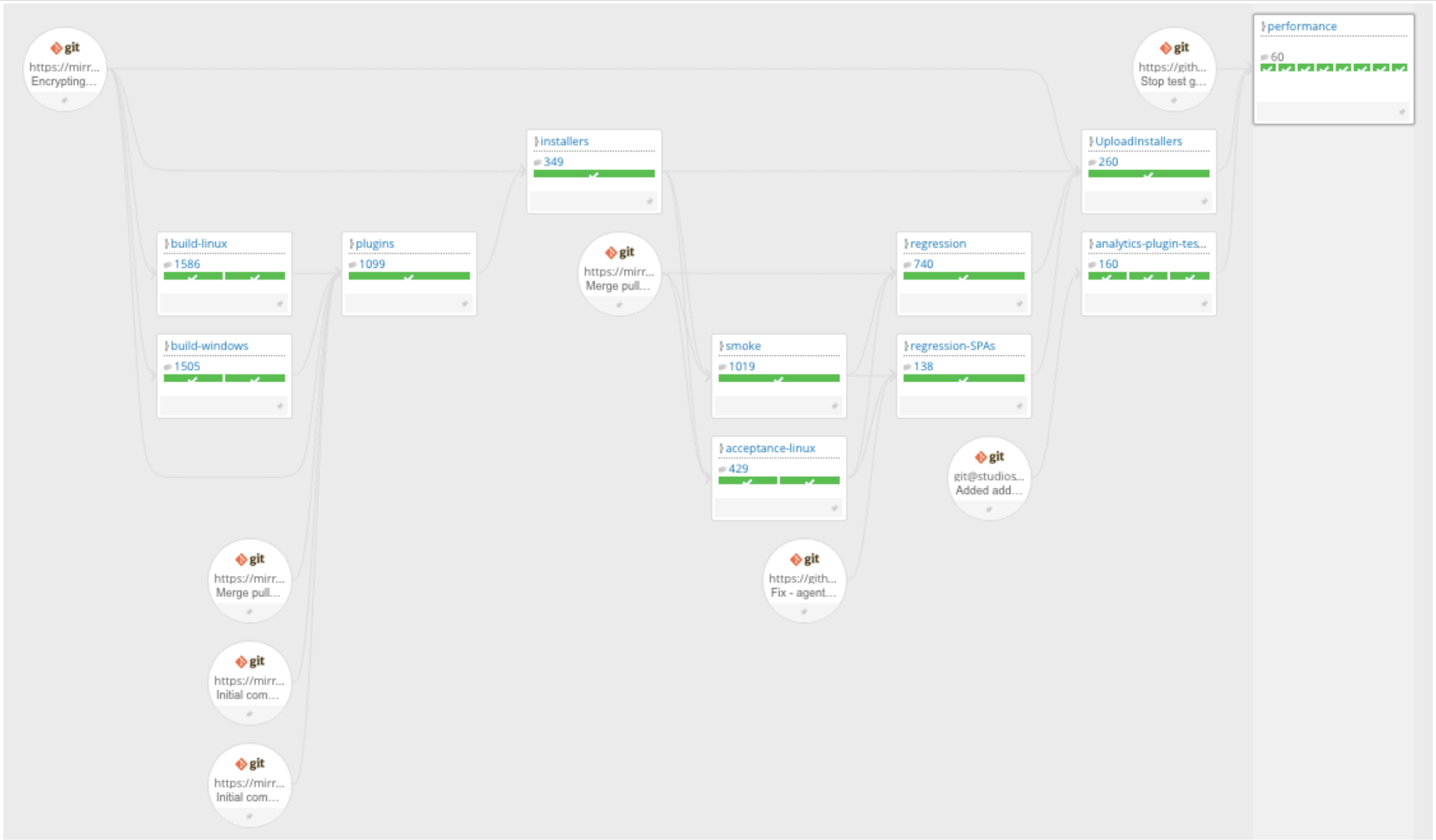
RUN AS MUCH AS POSSIBLE IN PARALLEL



$237,840

@kmugrage

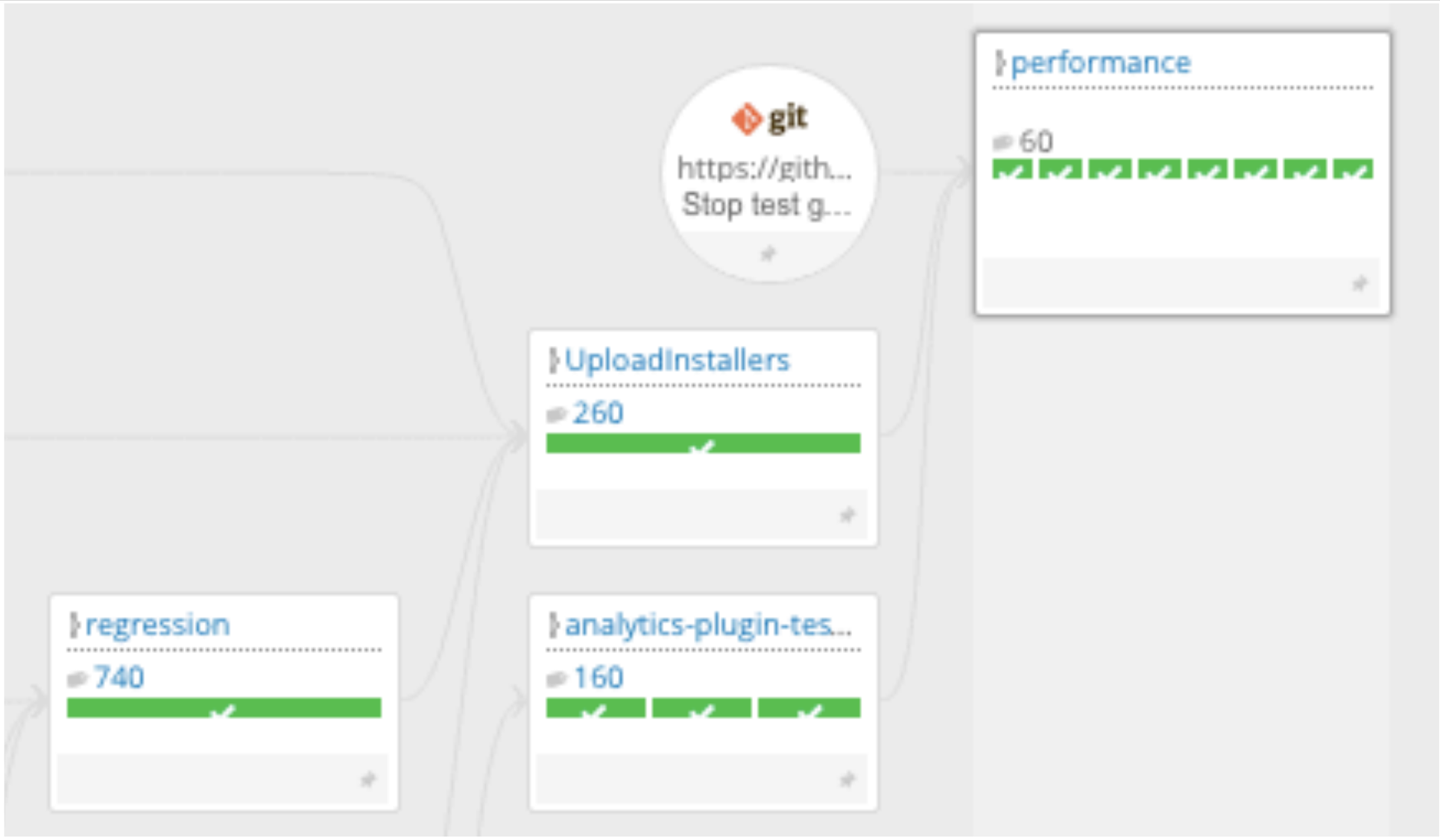
RUN AS MUCH AS POSSIBLE IN PARALLEL



$249,732

@kmugrage

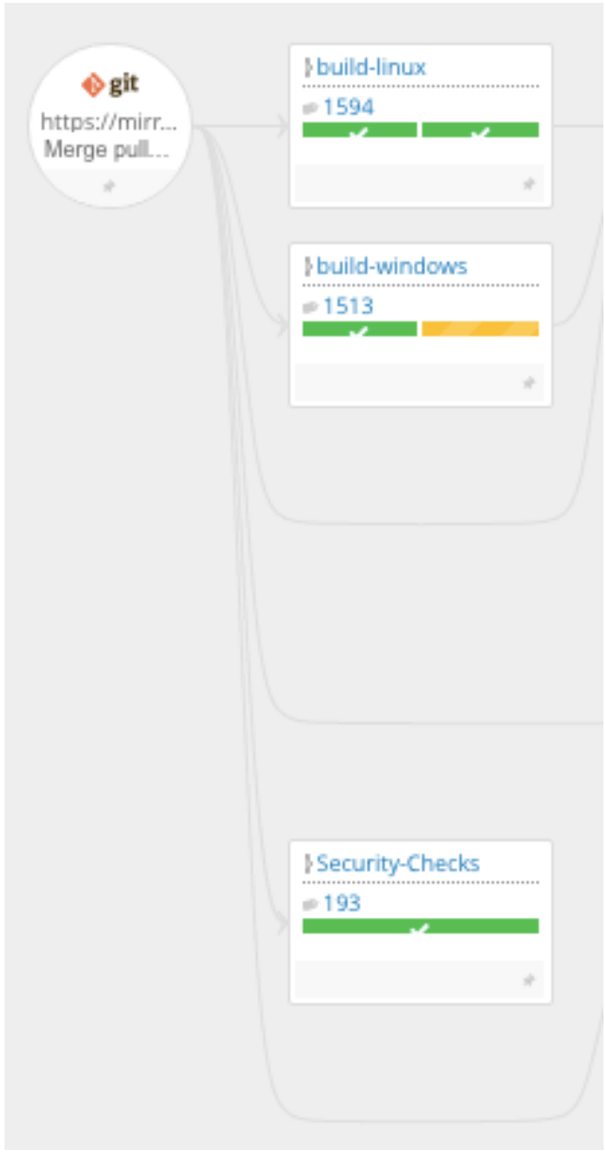
YOU SHOULD DECIDE THE ORDER, NOT YOUR TOOL



$261,624

@kmugrage

YOU SHOULD DECIDE THE ORDER, NOT YOUR TOOL



$273,516

@kmugrage

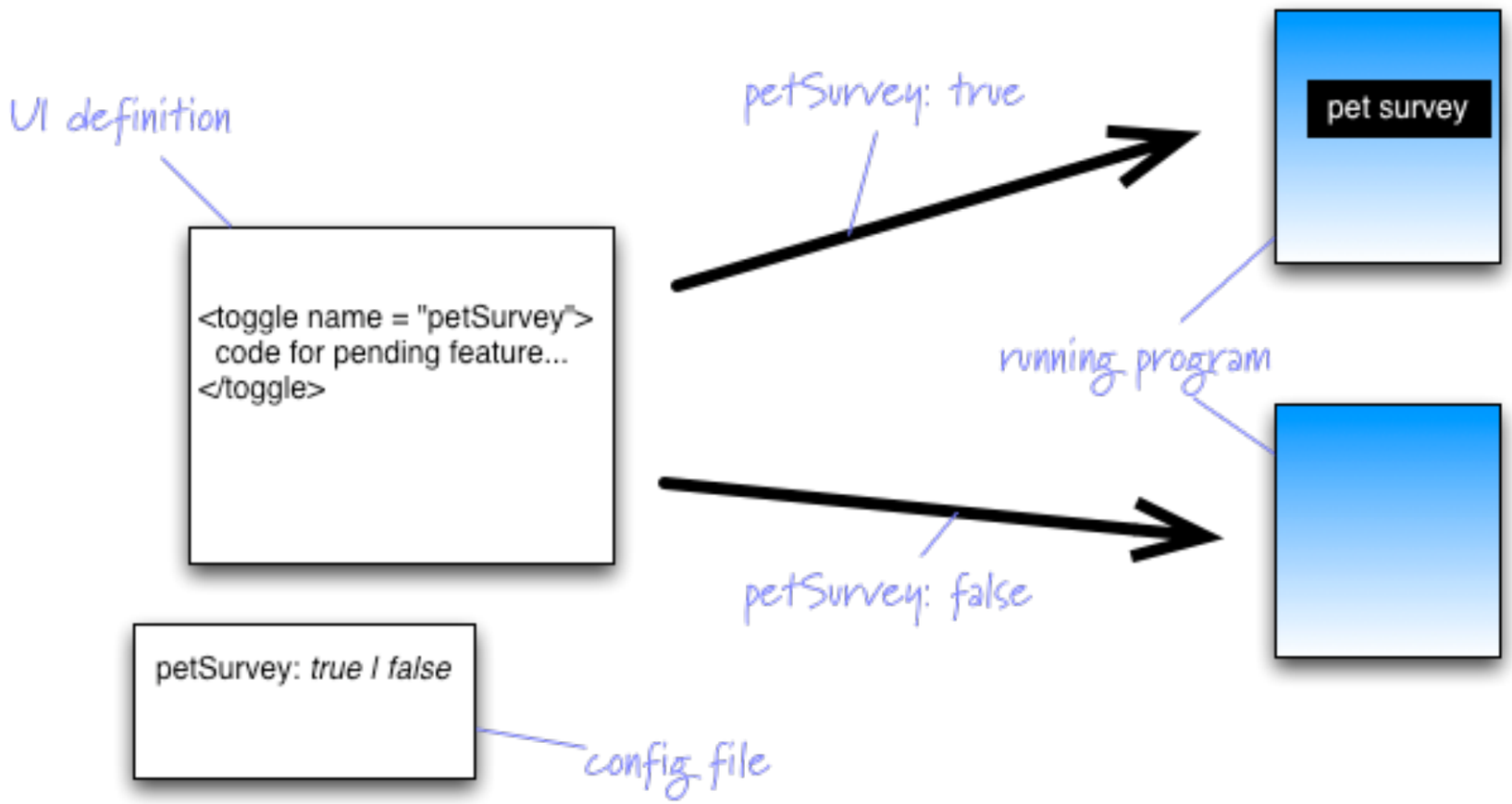
ThoughtWorks®

DEPLOYING INCOMPLETE WORK

How to deliver faster than you can finish a feature

$285,408

FEATURE TOGGLES

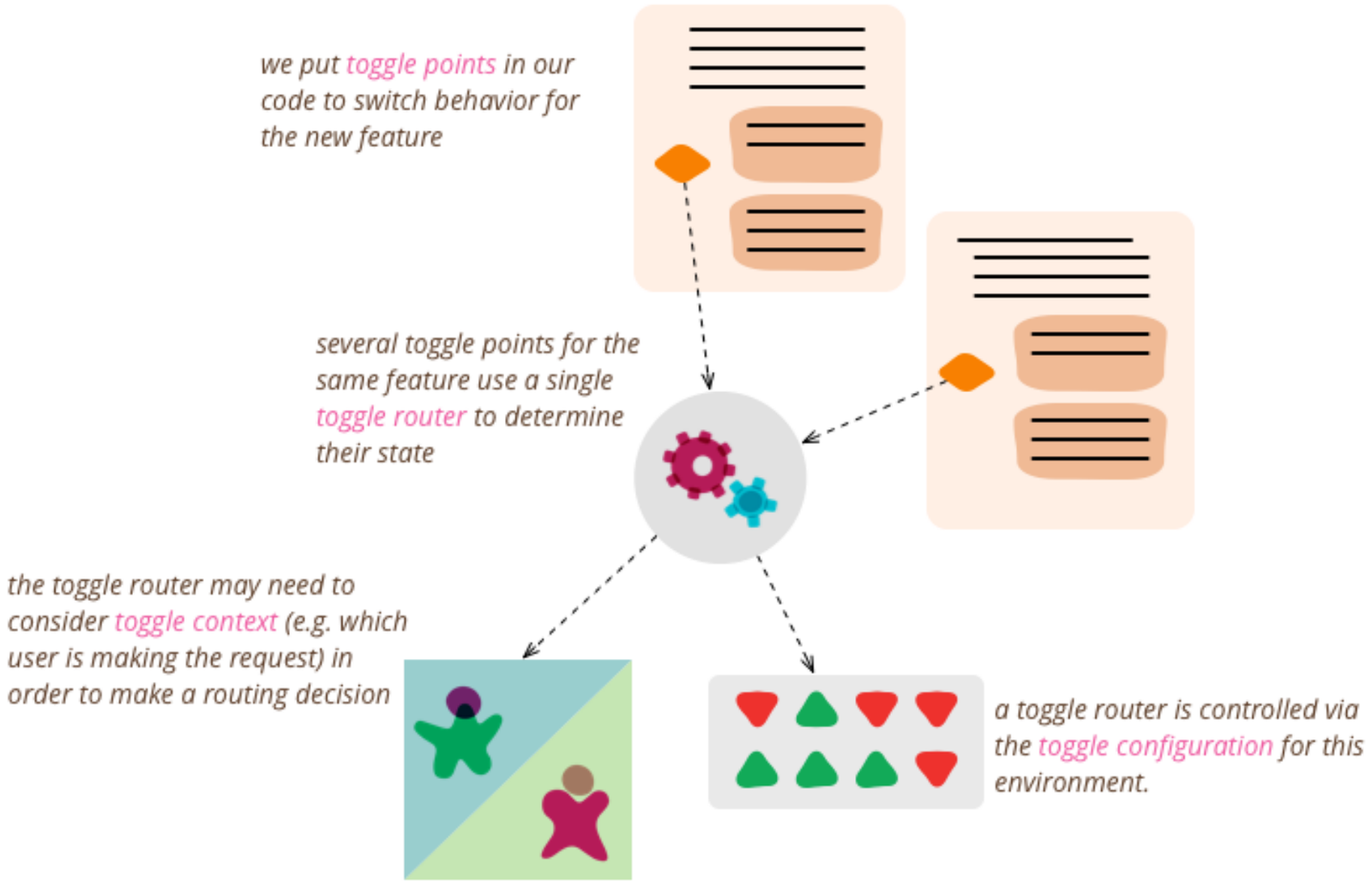


$297,300

https://martinfowler.com/bliki/FeatureToggle.html

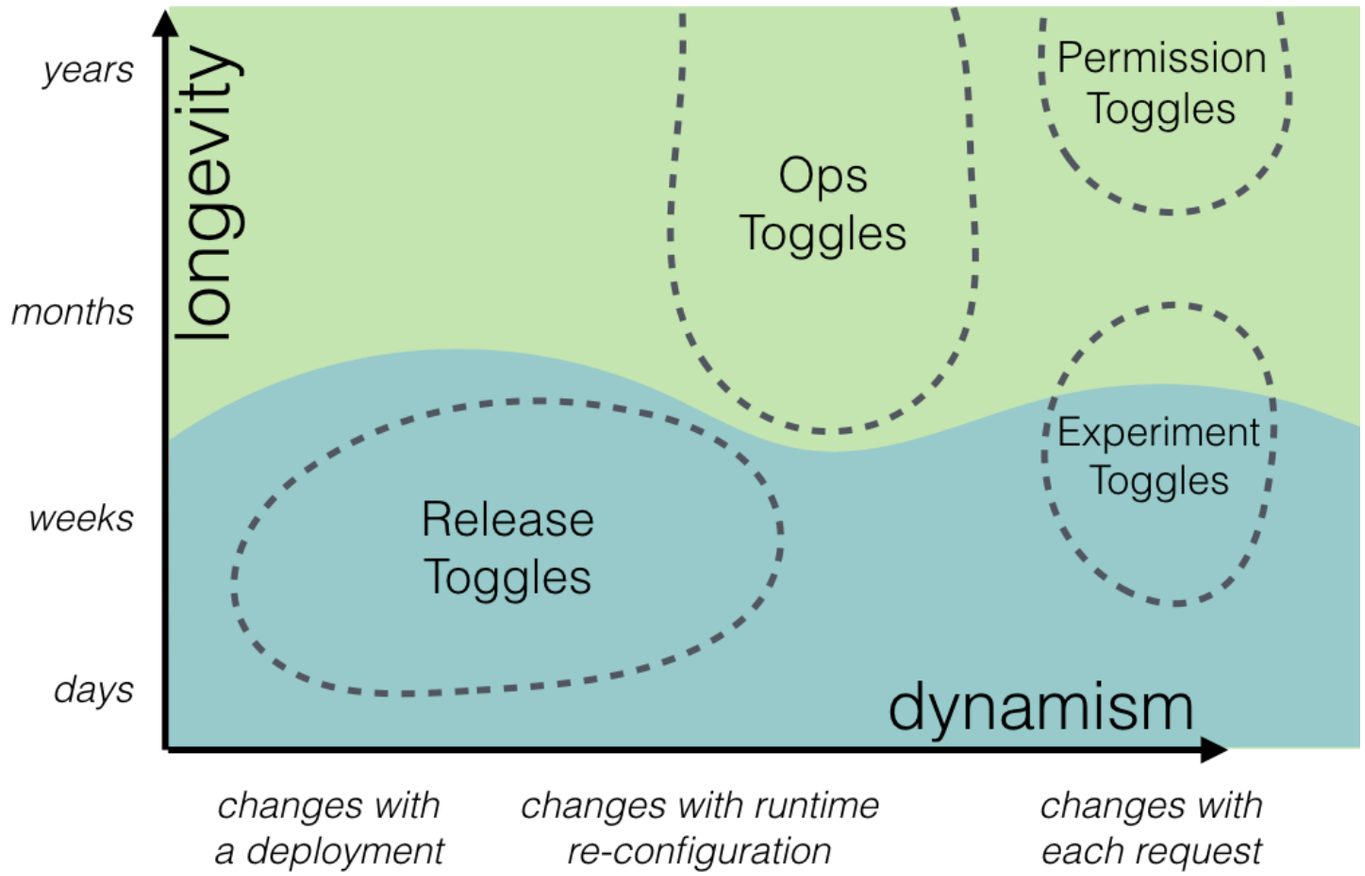
@kmugrage

FEATURE TOGGLES



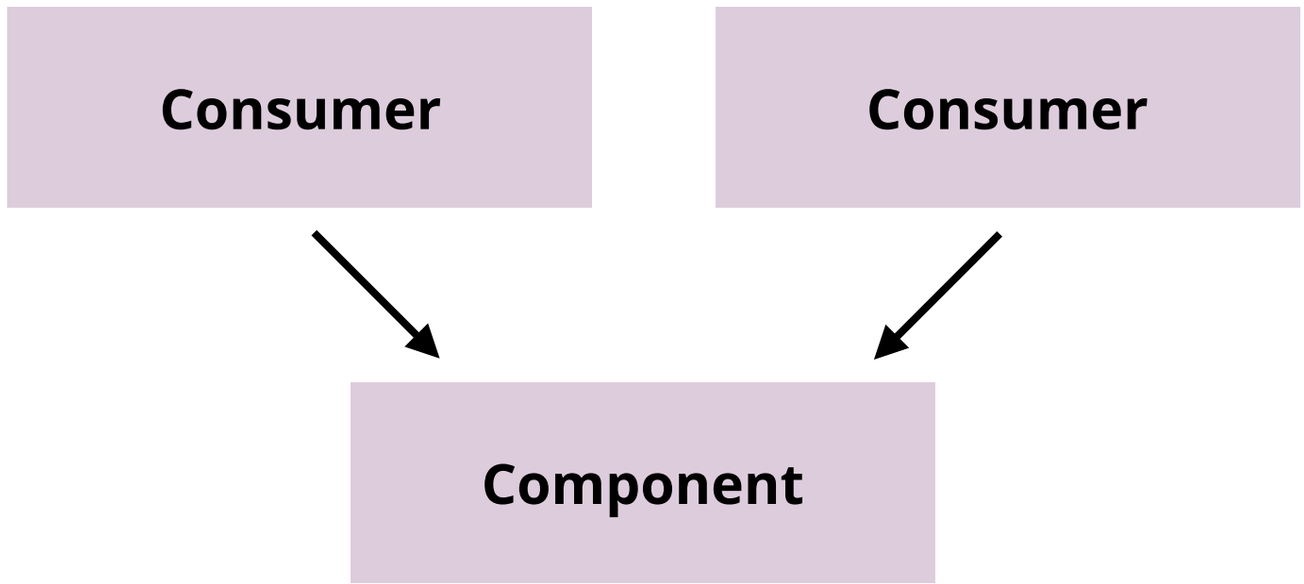
Pete Hodgson - http://martinfowler.com/articles/feature-toggles.html

FEATURE TOGGLES



Pete Hodgson - http://martinfowler.com/articles/feature-toggles.html

BRANCH BY ABSTRACTION

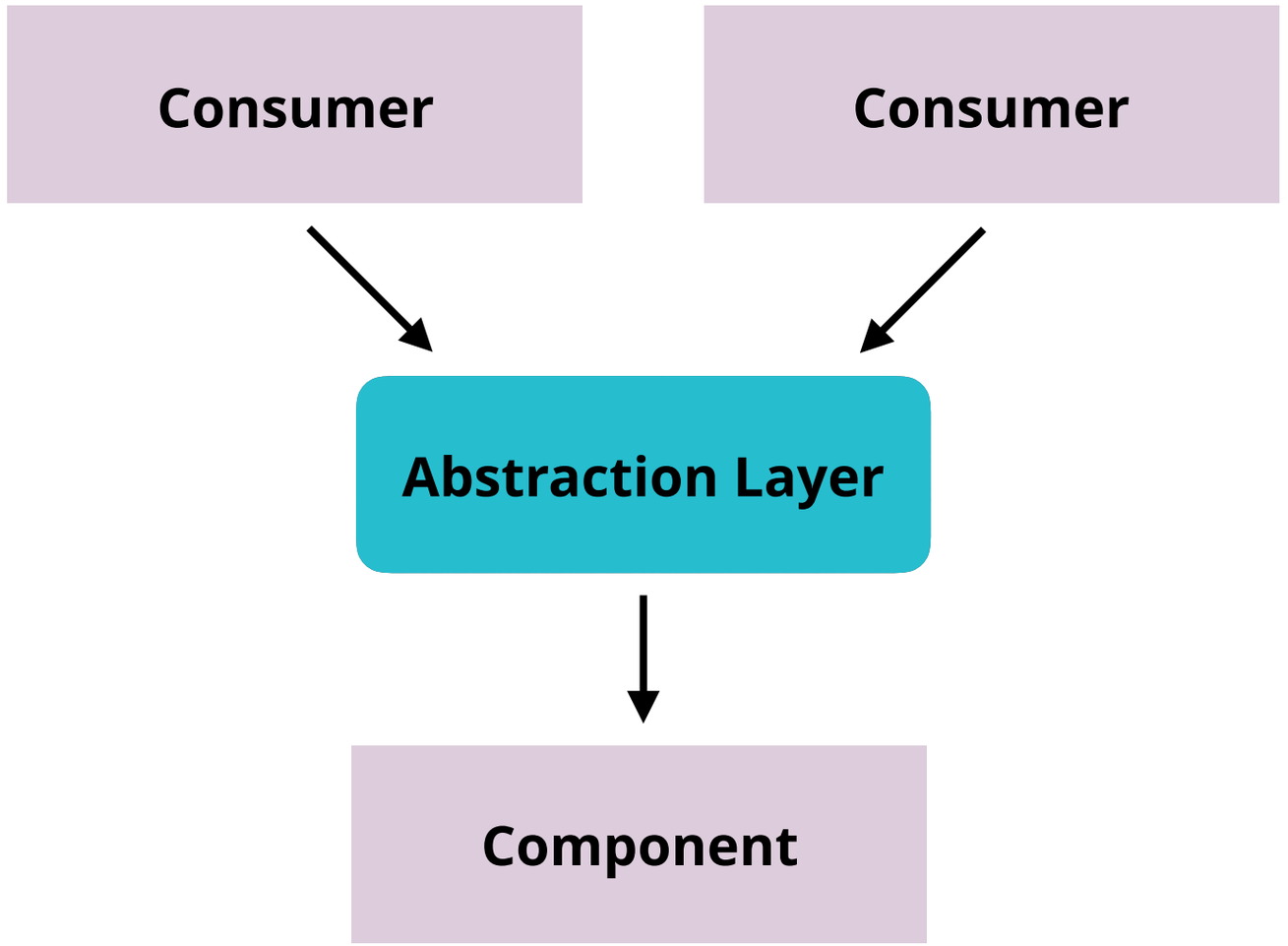


$332,976

@kmugrage

https://continuousdelivery.com/2011/05/make-large-scale-changes-incrementally-with-branch-by-abstraction/

BRANCH BY ABSTRACTION

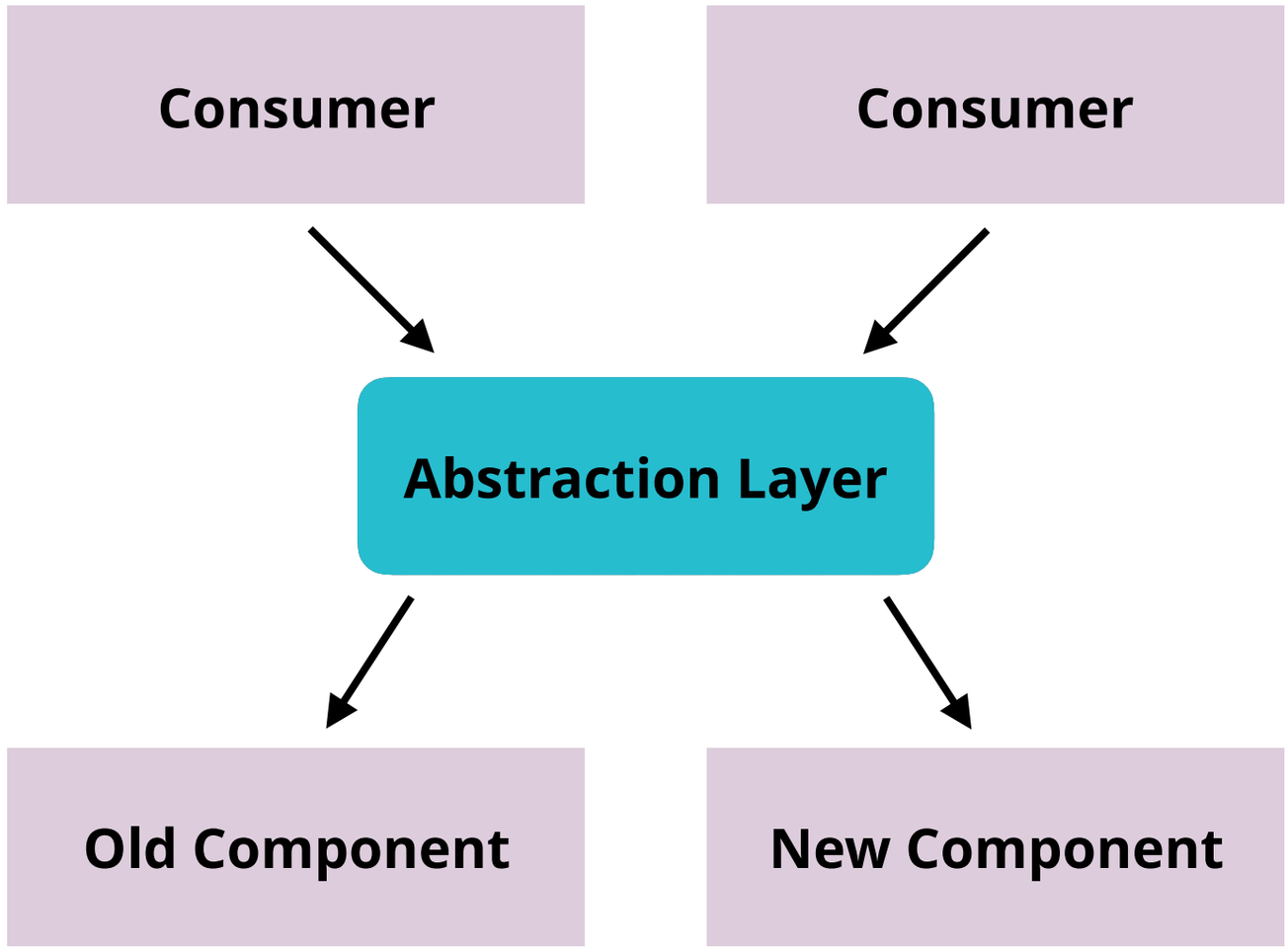


$344,868

@kmugrage

https://continuousdelivery.com/2011/05/make-large-scale-changes-incrementally-with-branch-by-abstraction/

BRANCH BY ABSTRACTION



$356,760

@kmugrage

https://continuousdelivery.com/2011/05/make-large-scale-changes-incrementally-with-branch-by-abstraction/

ThoughtWorks®

MANAGING RISK

$368,652

OPTIMIZE FOR RECOVERY

Mean time between failures (MTBF) is the predicted elapsed time between inherent failures of a system during operation.

Mean Time To Repair (MTTR) is a basic measure of the maintainability of repairable items. It represents the average time required to repair a failed component or device.

https://en.wikipedia.org/wiki/Mean_time_between_failures
https://en.wikipedia.org/wiki/Mean_time_to_repair

DEPLOYMENT PATTERNS

Canary release

A technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.

Dark launching

The practice of deploying the very first version of a service into its production environment, well before release, so that you can soak test it and find any bugs before you make its functionality available to users.

http://martinfowler.com/bliki/CanaryRelease.html

$392,436 *http://www.informit.com/articles/article.aspx?p=1833567&seqNum=2* **@kmugrage**

DEPLOYMENT PATTERNS

GitHub

https://githubengineering.com/move-fast/

$404,328

@kmugrage

FEEDBACK LOOPS

Create useful logging for everything

Run (some of) your tests against production

Ensure alerts are useful as well

KNIGHT CAPITAL

On August 1, 2012, Knight Capital deployed untested software to a production environment which contained an obsolete function. The incident happened due to a technician forgetting to copy the new Retail Liquidity Program (RLP)....

...sent millions of child orders, resulting in 4 million executions in 154 stocks for more than 397 million shares in approximately 45 minutes.

Knight Capital took a pre-tax loss of

$440,000,000

@kmugrage

KNIGHT CAPITAL

On August 1, 2012, Knight Capital deployed untested software to a production environment which contained an obsolete function. The incident happened due to a technician forgetting to copy the new Retail Liquidity Program (RLP)....

...sent millions of child orders, resulting in 4 million executions in 154 stocks for more than 397 million shares in approximately 45 minutes.

Knight Capital took a pre-tax loss of ***$440,000,000***

@kmugrage

SUMMARY

It’s not Continuous Delivery if you can’t deploy right now

Practice good CI habits

Use things like feature toggles to deploy incomplete work

@kmugrage