

ThoughtWorks®

IT'S NOT CONTINUOUS DELIVERY

If you can't deploy to production right now

WHO AM I?

Ken Mugrage

ThoughtWorks Technology Evangelist

@kmugrage

THE THOUGHTWORKS STORY



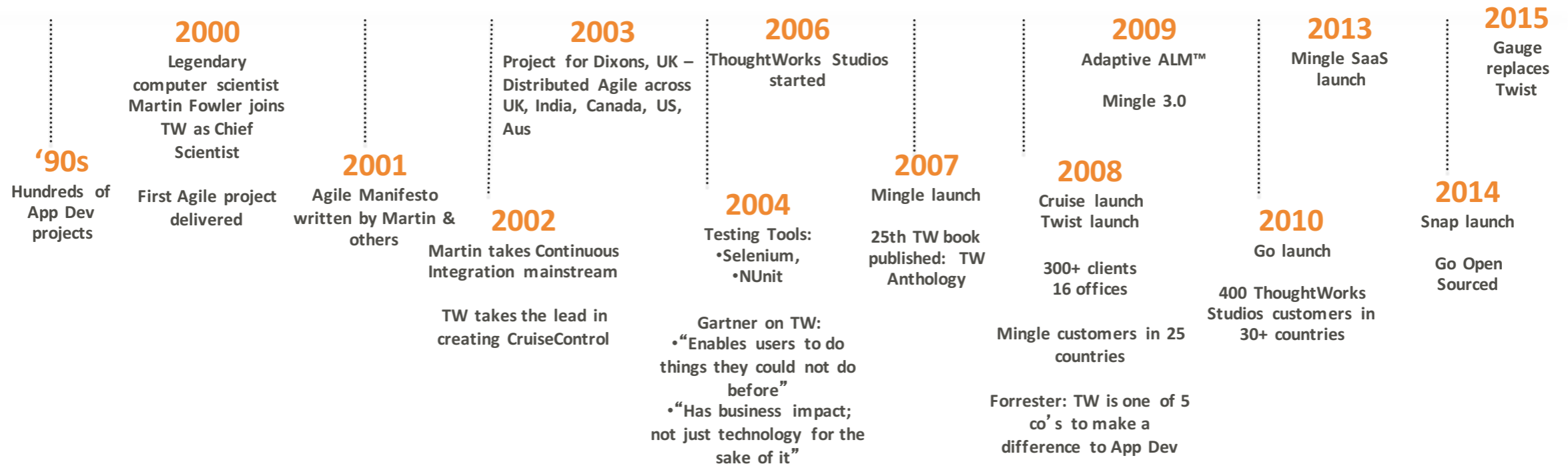
ThoughtWorks®
STUDIOS



mingle®



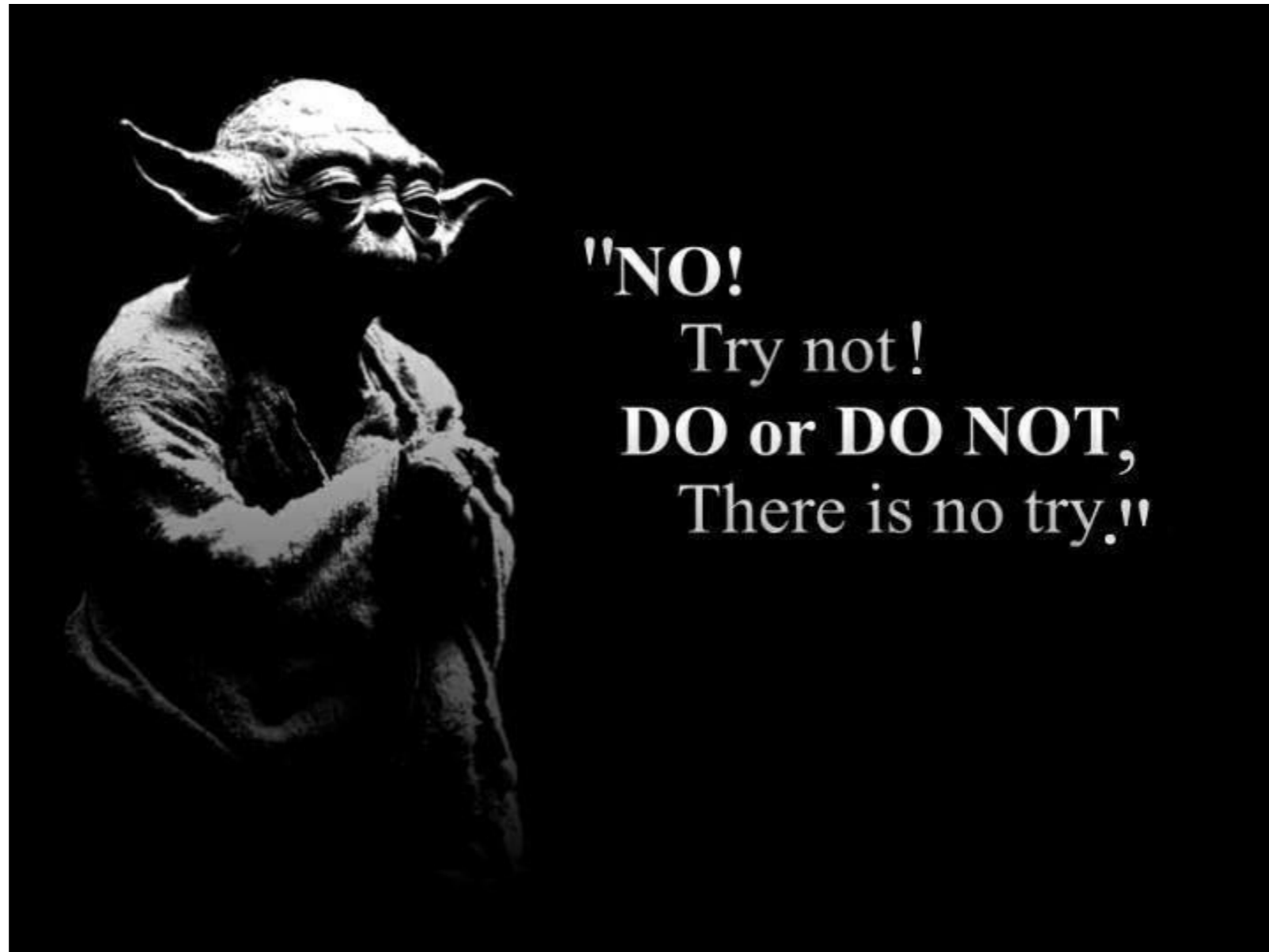
go™



ThoughtWorks®

WHY THIS TALK

THERE IS NO TRY



WHY CONTINUOUS DELIVERY?



PARTIALLY “DONE” MIGHT STILL BE USEFUL

PARTIALLY "DONE" MIGHT STILL BE USEFUL



1



2



3



PARTIALLY "DONE" MIGHT STILL BE USEFUL



1



2



3



1



2



3



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.

The Electronic Frontier Foundation, Ars Technica, and Bruce Schneier all deemed the Heartbleed bug "catastrophic". Forbes cybersecurity columnist Joseph Steinberg wrote, "Some might argue that [Heartbleed] is the worst vulnerability found (at least in terms of its potential impact) since commercial traffic began to flow on the Internet.

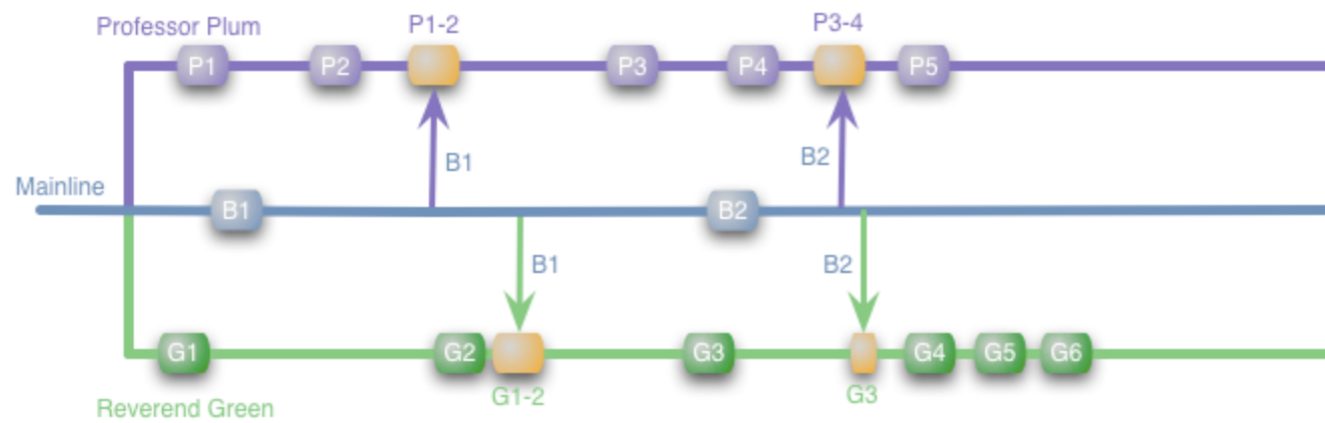
<https://en.wikipedia.org/wiki/Heartbleed>



CODE MANAGEMENT

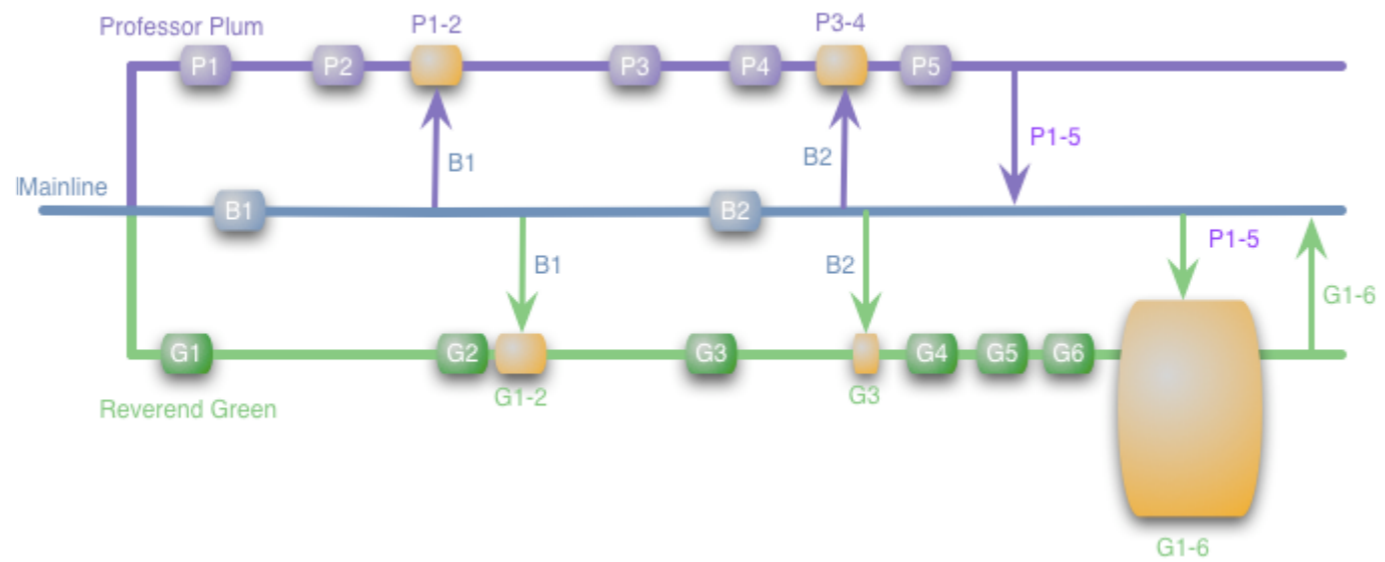
Working with your version control system

FEATURE BRANCHING



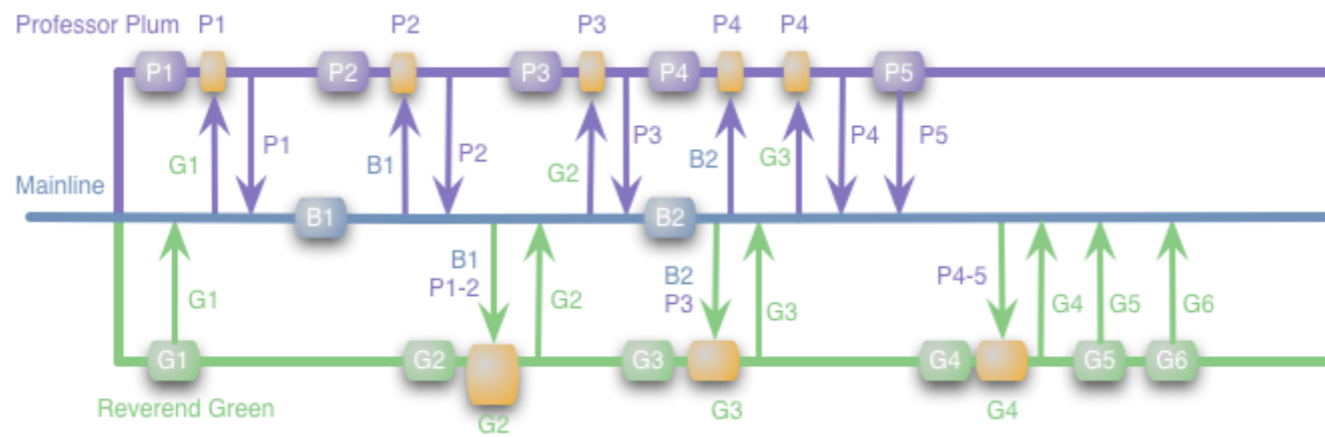
<http://martinfowler.com/bliki/FeatureBranch.html>

FEATURE BRANCHING



<http://martinfowler.com/bliki/FeatureBranch.html>

FEATURE BRANCHING



<http://martinfowler.com/bliki/FeatureBranch.html>

CODE MANAGEMENT

Recommended CI practices

CODE MANAGEMENT

Recommended CI practices

Everyone commits to trunk at least daily

CODE MANAGEMENT

Recommended CI practices

Everyone commits to trunk at least daily

Automated tests are run for every commit

CODE MANAGEMENT

Recommended CI practices

Everyone commits to trunk at least daily

Automated tests are run for every commit

Avoid branches

RELEASING INCOMPLETE WORK

How to deliver faster than you can finish a feature

FEATURE TOGGLES

FEATURE TOGGLES

before

```
function calculateTax(){  
  // current implementation lives here  
}
```

FEATURE TOGGLES

before

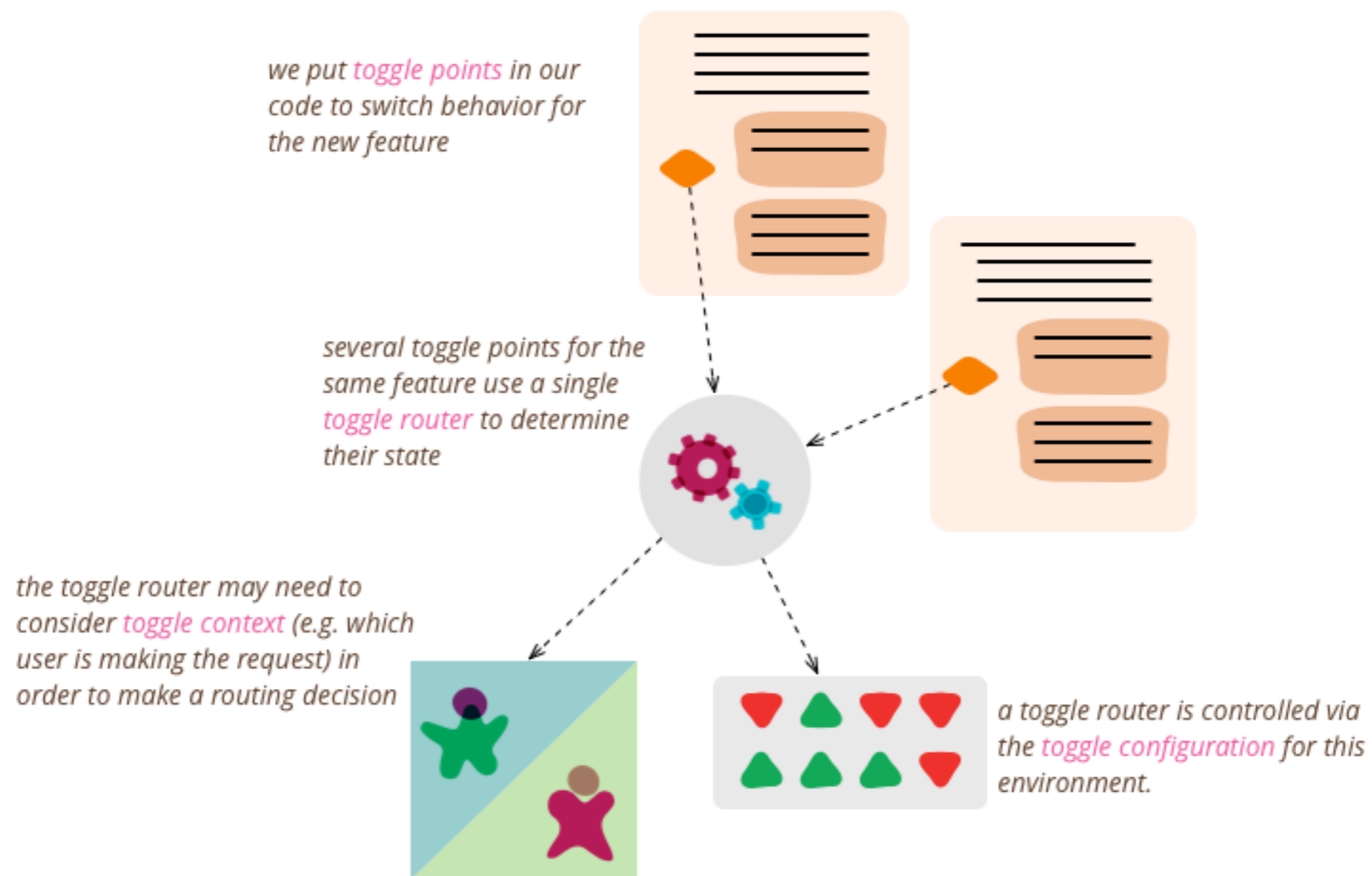
```
function calculateTax(){  
  // current implementation lives here  
}
```

after

```
function calculateTax(){  
  var useNewAlgorithm = false;  
  // useNewAlgorithm = true; // UNCOMMENT IF YOU ARE WORKING ON THE NEW SR ALGORITHM  
  
  if( useNewAlgorithm ){  
    return enhancedSplineReticulation();  
  }else{  
    return oldFashionedSplineReticulation();  
  }  
}  
  
function oldFashionedTaxCalculation(){  
  // current implementation lives here  
}  
  
function enhancedTaxCalculation(){  
  // TODO: implement better SR algorithm  
}
```

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

FEATURE TOGGLES



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

ThoughtWorks®

**PIPELINES YOU
SHOULD BE
INCLUDING**

SECURITY TESTING

SECURITY TESTING

Test before you commit

Have you included private keys? Authentication tokens?

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.”

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.

PERFORMANCE TESTING

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

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.

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

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.

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

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.

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

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.

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

PIPELINE STRATEGIES

Recommended practices

PIPELINE STRATEGIES

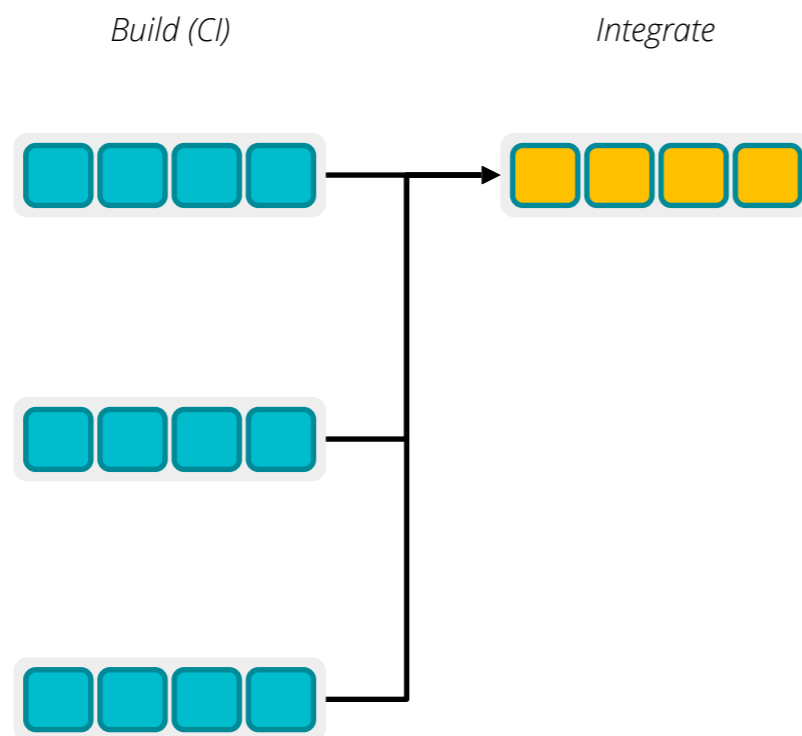
Recommended practices

Build (CI)



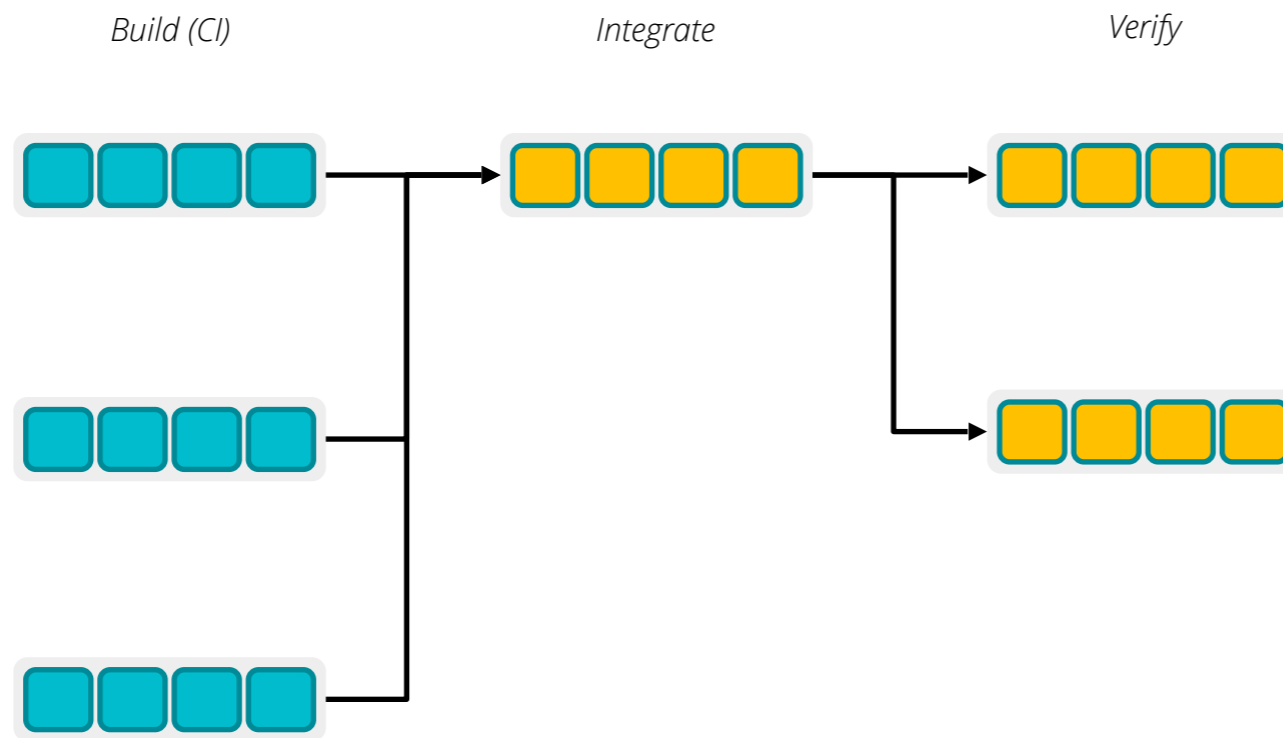
PIPELINE STRATEGIES

Recommended practices



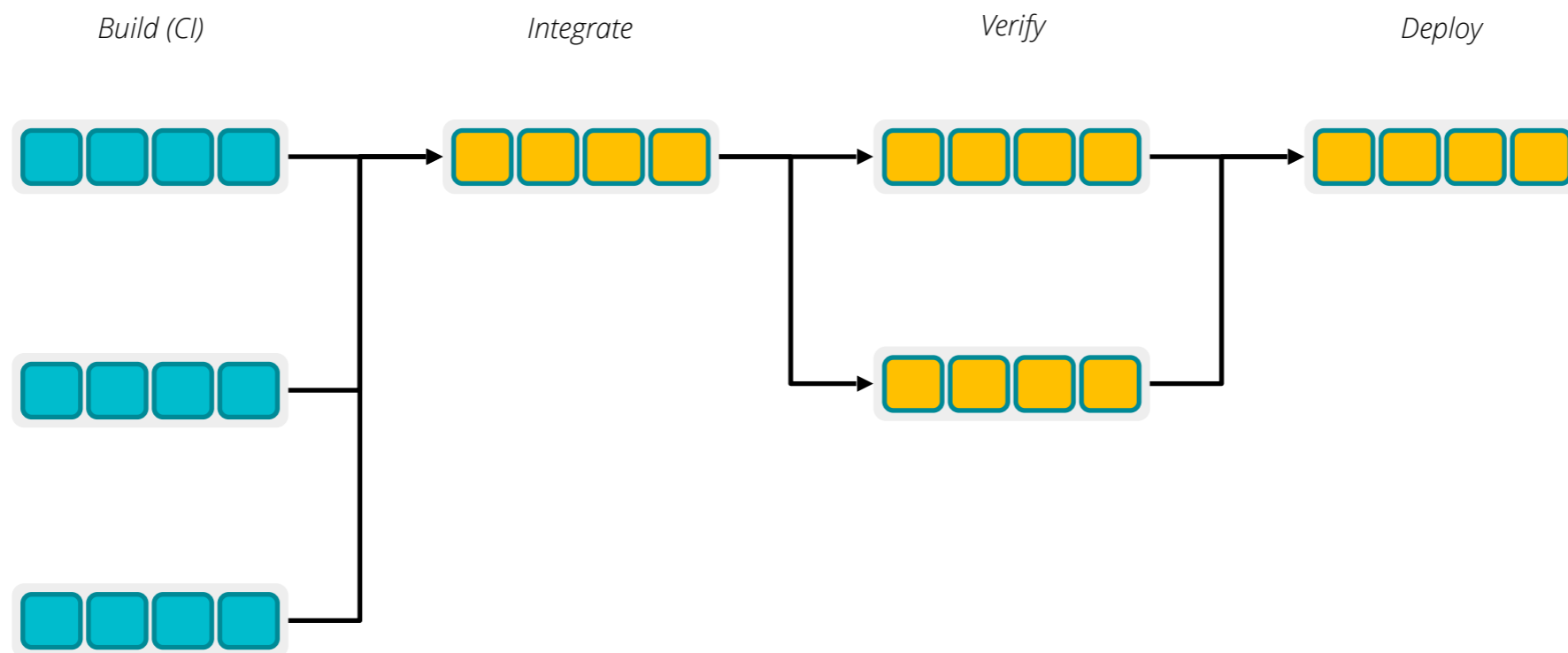
PIPELINE STRATEGIES

Recommended practices

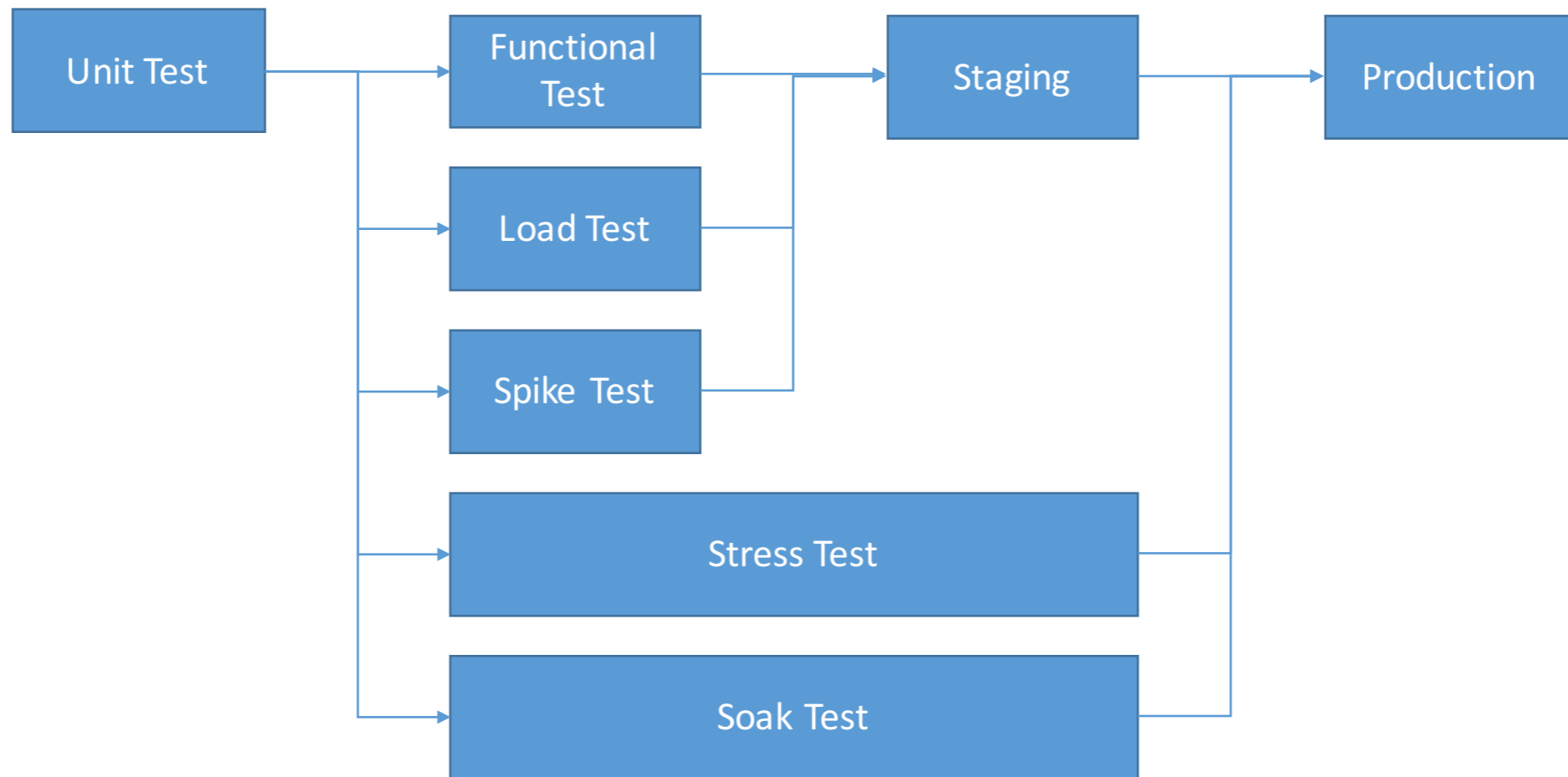


PIPELINE STRATEGIES

Recommended practices



RUN AS MUCH AS POSSIBLE IN PARALLEL



MANAGING RISK

DEPLOYMENT PATTERNS

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.

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>

<http://www.informit.com/articles/article.aspx?p=1833567&seqNum=2>

FEEDBACK LOOPS

FEEDBACK LOOPS

Create useful logging for everything

FEEDBACK LOOPS

Create useful logging for everything

Run (some of) your tests against production

FEEDBACK LOOPS

Create useful logging for everything

Run (some of) your tests against production

Configure monitoring alerts to make sure people pay attention to them

OPTIMIZE FOR RECOVERY

OPTIMIZE FOR RECOVERY

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

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

SUMMARY

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

Practice good CI habits

Use things like feature branches to deploy incomplete work

ThoughtWorks®

THANK YOU

To learn more about ThoughtWorks CD Products
<http://www.thoughtworks.com/products/>