What is DevOps?

What is DevOps?

DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.

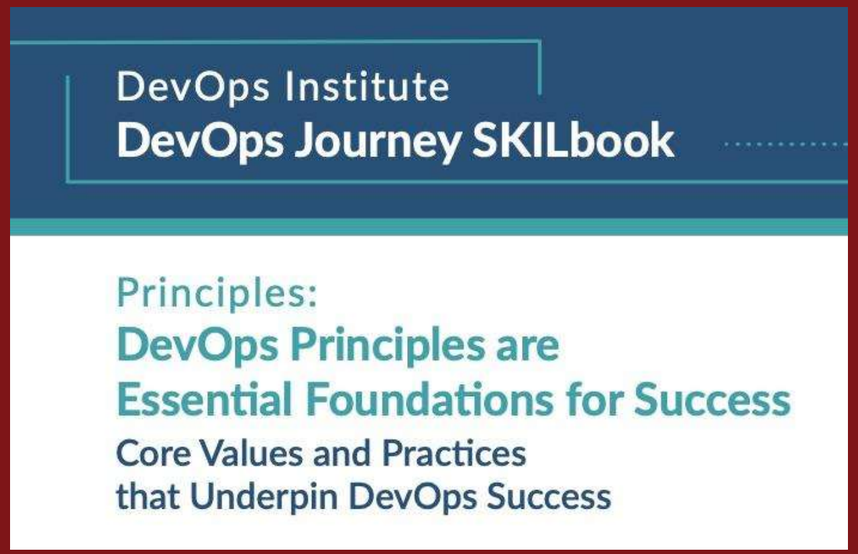
The contraction of “Dev” and “Ops” refers to replacing siloed Development and Operations to create multidisciplinary teams that now work together with shared and efficient practices and tools.

- ***Culture***
- ***Automation***
- ***Lean***
- ***Measurement***
- ***Sharing***

What is DevOps & What is not DevOps?

What DevOps is?	What DevOps is not?
• A practice	• A permanent team
• A culture & mindset	• A profession
• Collaboration	• Just software tool
• Agile operations	• Just code infrastructure
• Software approach for operations	• A system engineer
• Fast IT service delivery	• Standard IT service management

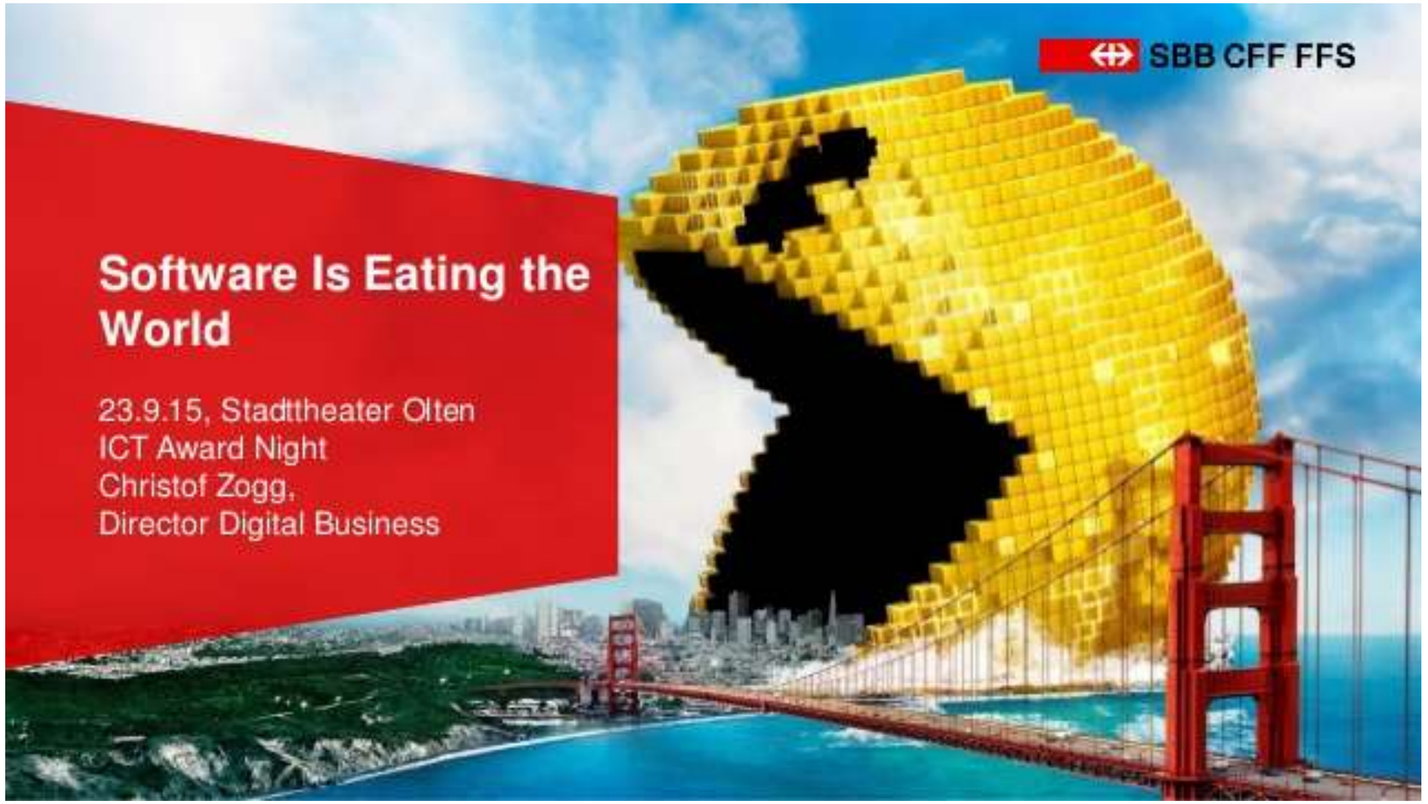
What is DevOps



"Imagine a world where product owners, Development, QA, IT Operations and InfoSec work together, not only to help each other, but also to ensure that the overall organization succeeds.

By working towards a common goal, they enable the fast flow of planned work into production, while achieving world-class stability, reliability, availability and security."

Why DevOps matter?



DevOps accelerates Digital Transformation

Quote from former CEO of Accenture at World Economic Forum

All Companies are now sharing their DevOps Stories

Web Pioneers

- Amazon
- Netflix
- Etsy
- Facebook
- Salesforce
- Google

Financial Services

- Capital One
- Bank of America
- Barclays
- Wells Fargo
- ING Bank
- UBS
- American Express

Entertainment & Media

- Disney
- Sony Pictures
- Hearst
- Verizon

Insurance

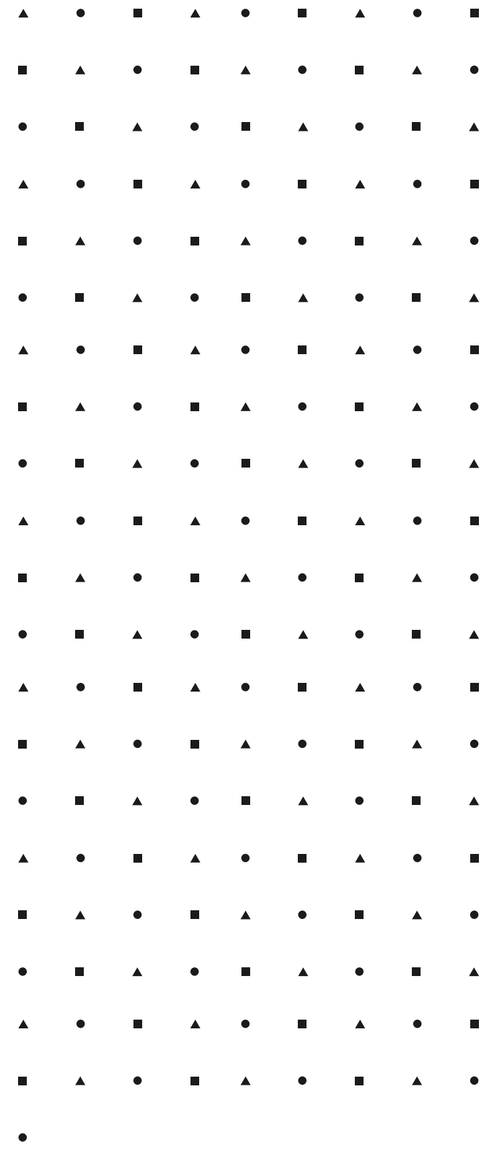
- Nationwide
- Travelers
- Kaiser Permanente
- Hiscox

Manufacturing

- Apple
- LEGO
- Unilever
- Jaguar Land Rover
- Adidas
- Nike

Retail

- Target
- Nordstrom
- Sherwin Williams
- Macy's
- Walmart
- Starbucks



Business Benefits of DevOps

Higher-quality

More reliable

Faster pace

Resulting in
Accelerated time to market

Improved experience for customers

Increased revenue

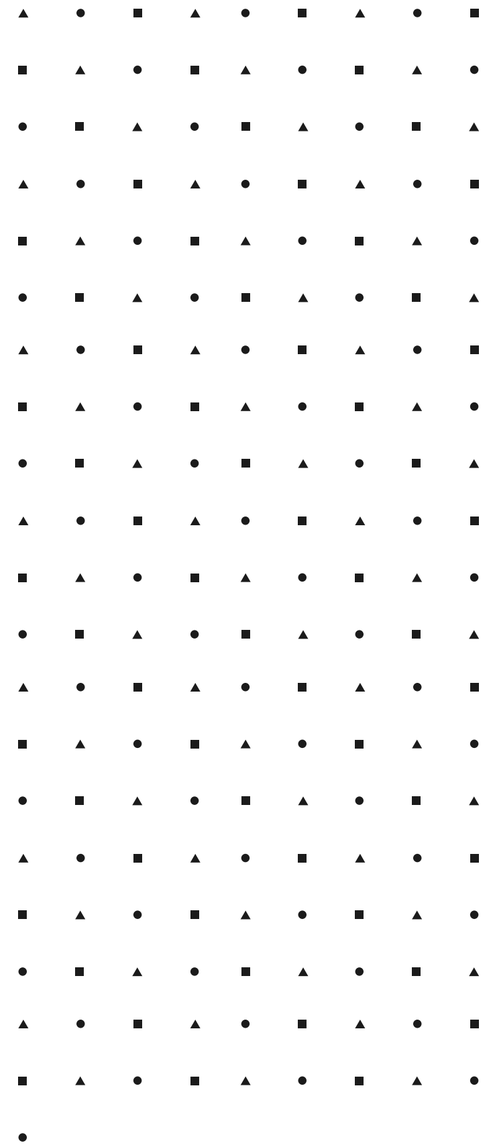
Enhanced collaboration and communication,

Faster development and delivery processes

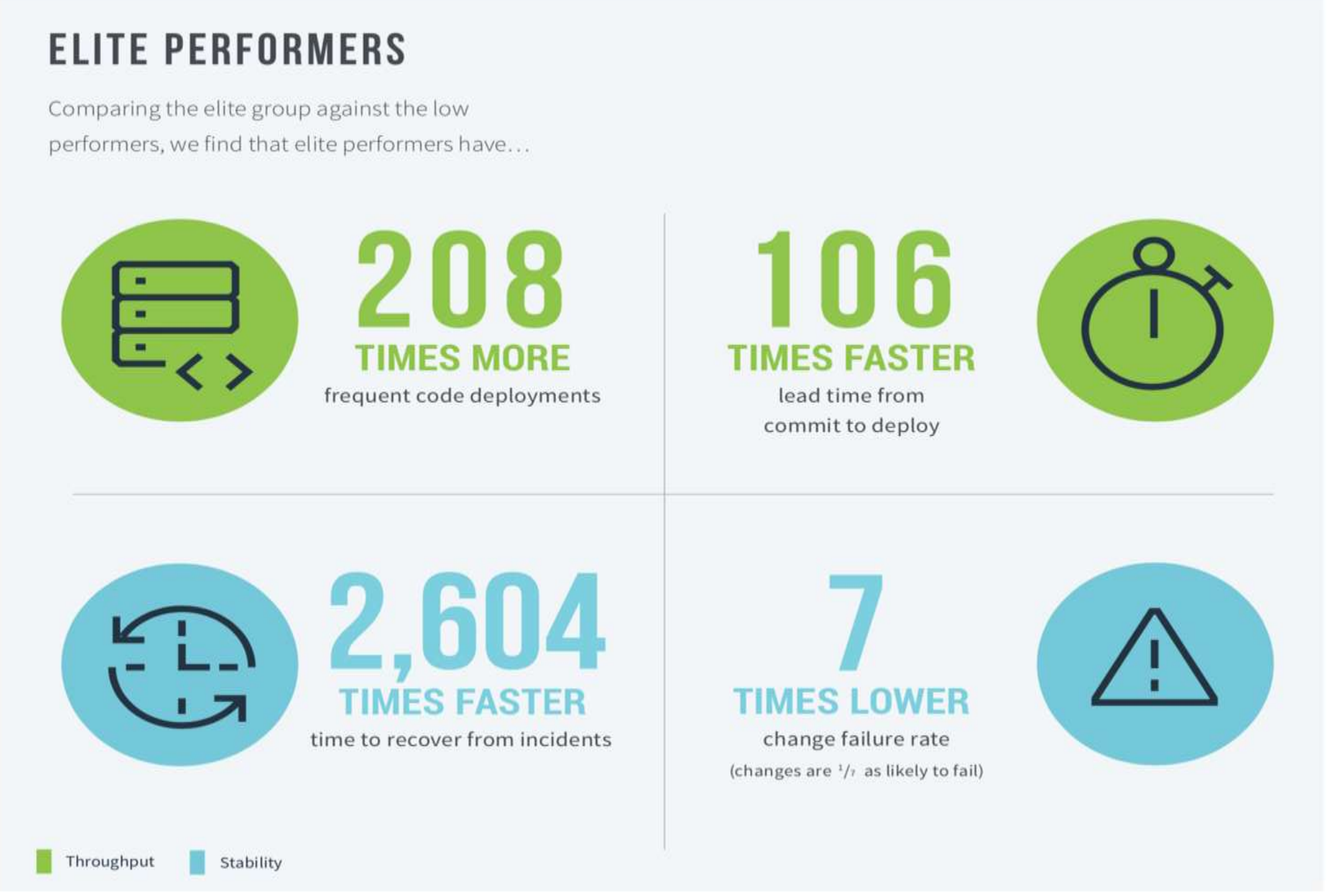
While delivering features, fixes and updates.

Bridges the communications gap between developer teams and IT operations

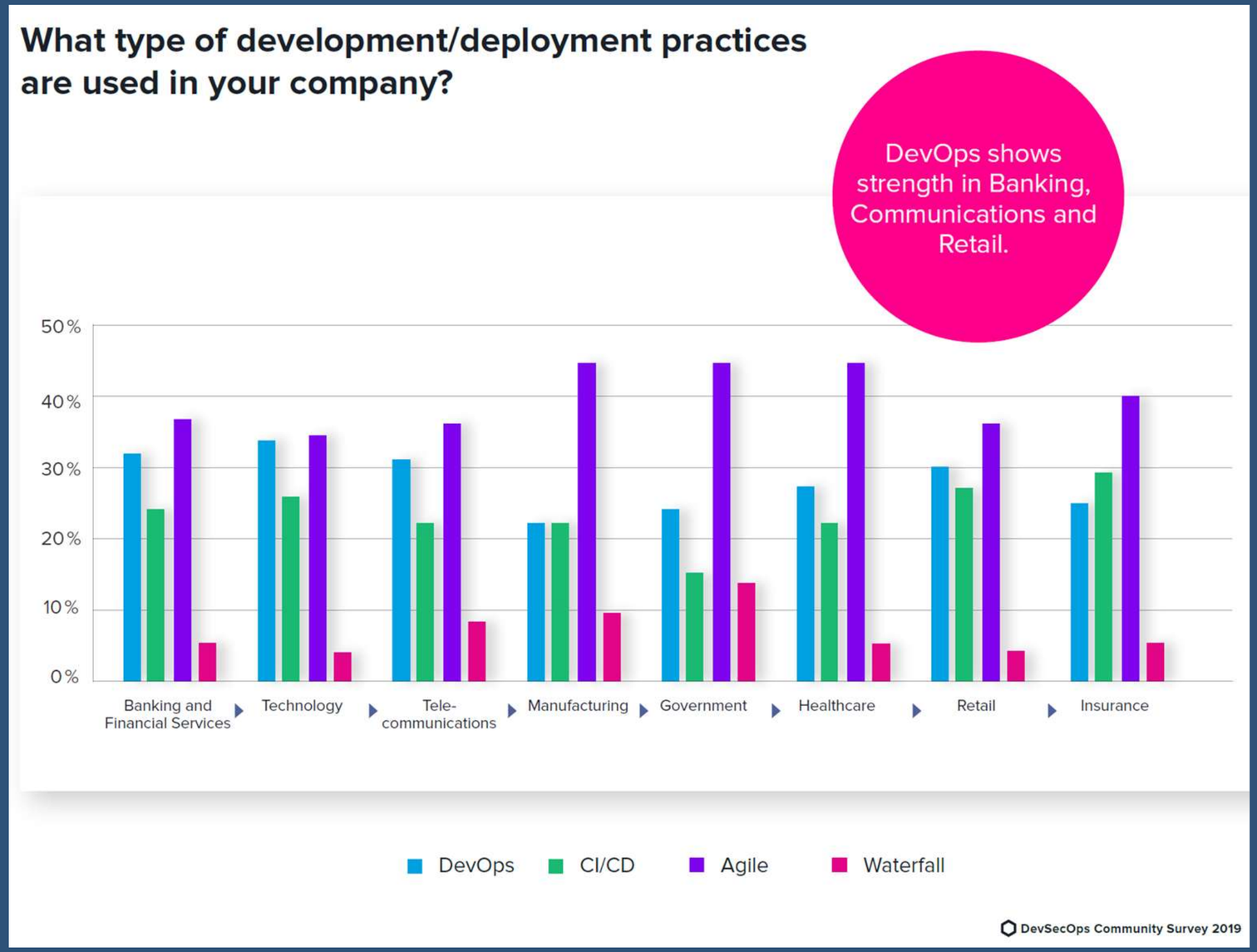
Quickly deliver solutions that best match the needs of both the customer and the business.



Business Benefits of DevOps



DevOps Adoption Across Sectors



Key Challenges for Organizations

Decentralized IT communities

Need to upskill Individuals

Organizations facing talent gap

Need to improve software delivery

Need to align to enterprise goals

You can help by gaining DevOps skills to overcome some of these challenges

Devops History Lesson

2008 Patrick Dubois is a consultant working on a DC migration project for the Belgian government

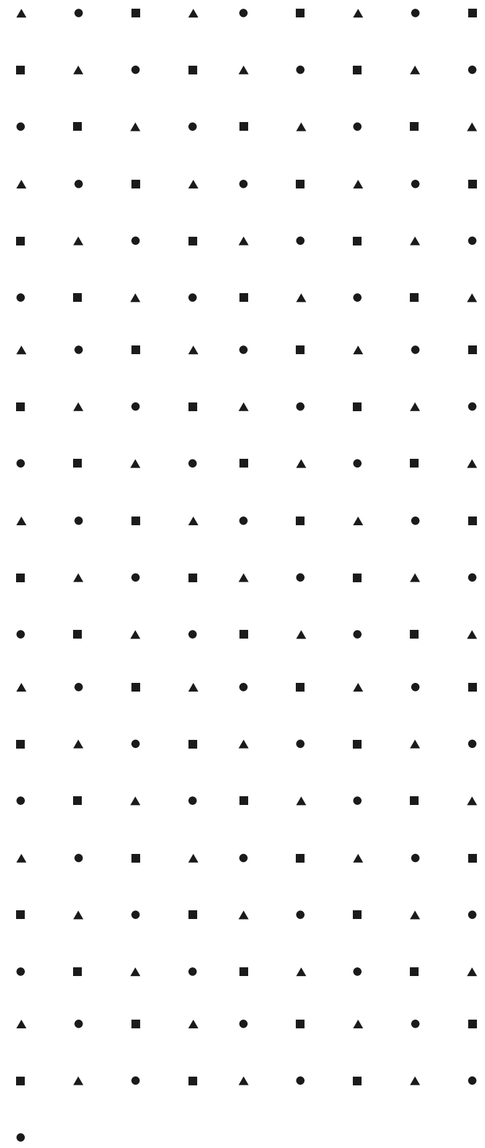
2008 Agile Systems Administration group formed by Dubois and Andrew Shafer

2009 Velocity ‘09 – John Allspaw & Paul Hammond give talk on 10+ Deploys a day at Flickr

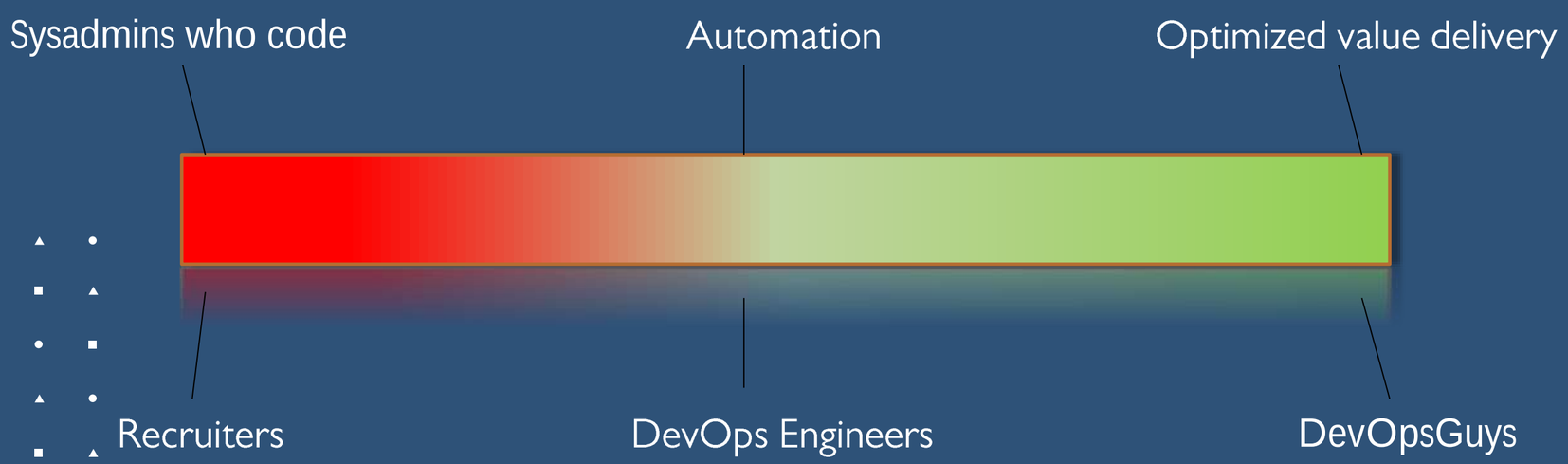
2009 First DevOpsDays conference

2010 First DevOpsDays US

2013 DevOps defined (sort of) as “an intimate understanding between the development and operations teams”



Devops Defined



Key Challenges for Organizations

Decentralized IT communities

Need to upskill Individuals

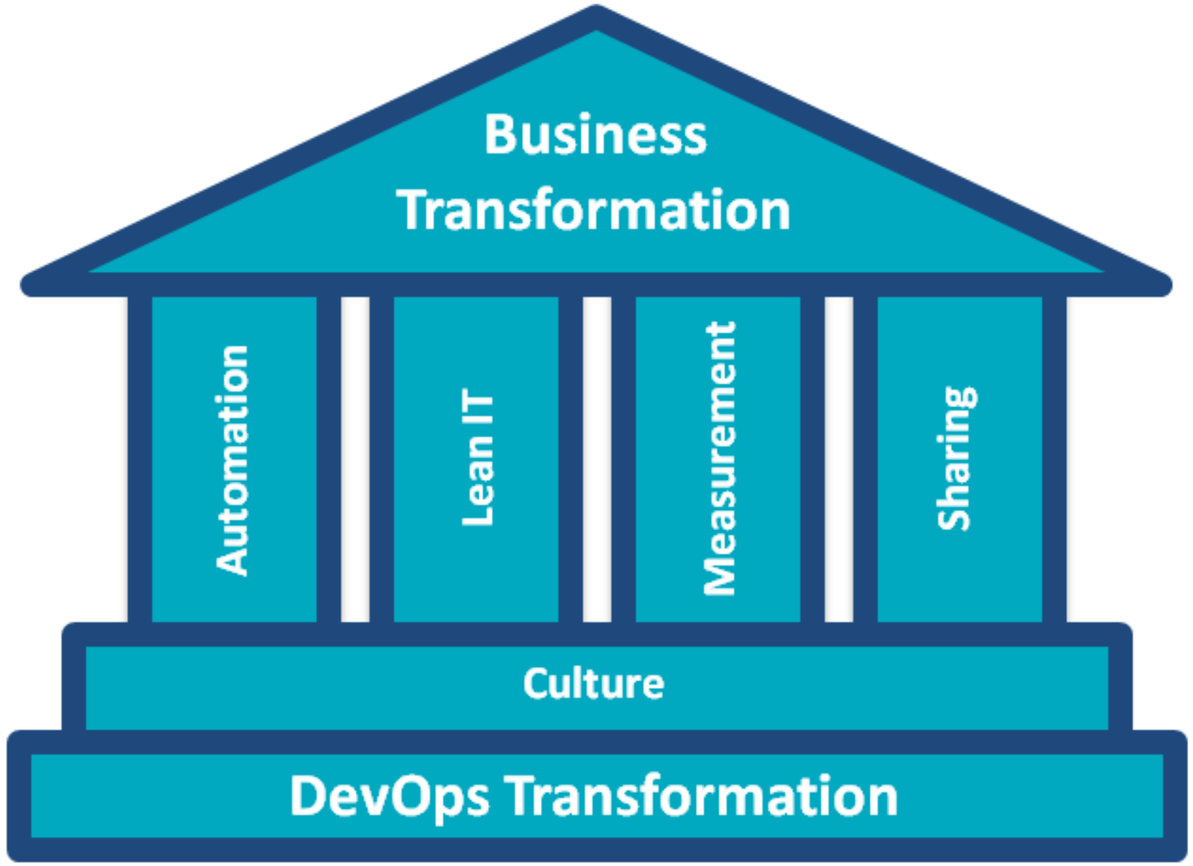
Organizations facing talent gap

Need to improve software delivery

Need to align to enterprise goals

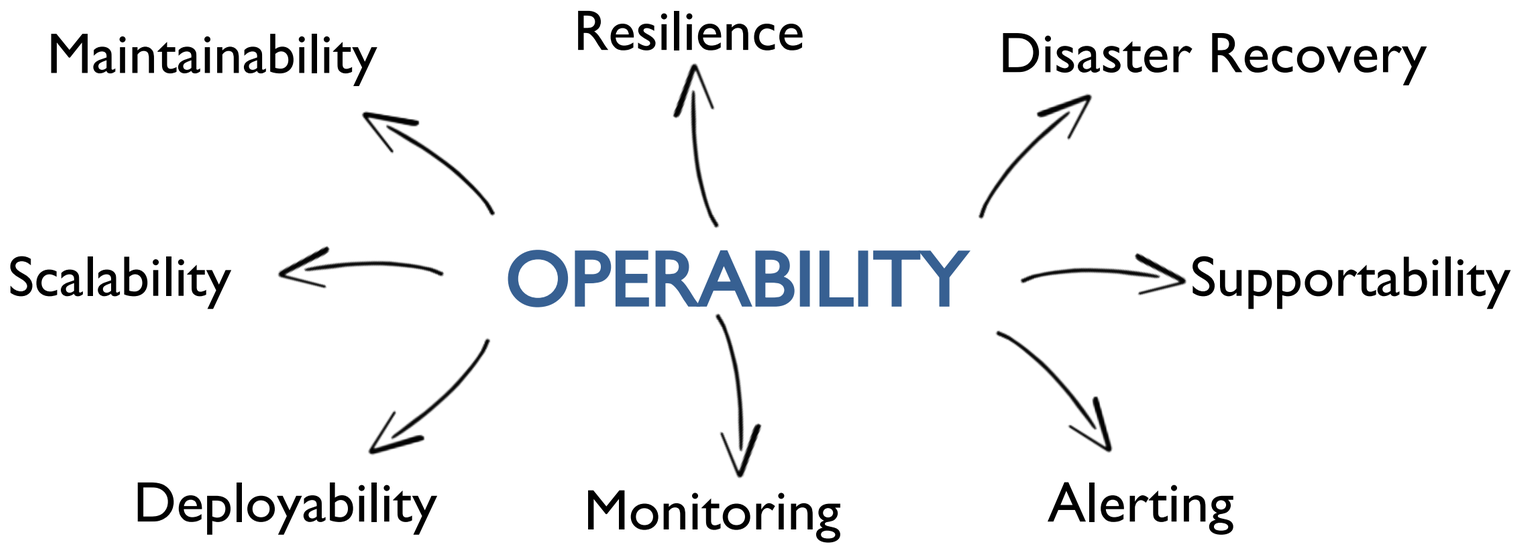
You can help by gaining DevOps skills to overcome some of these challenges

The CALMS Model

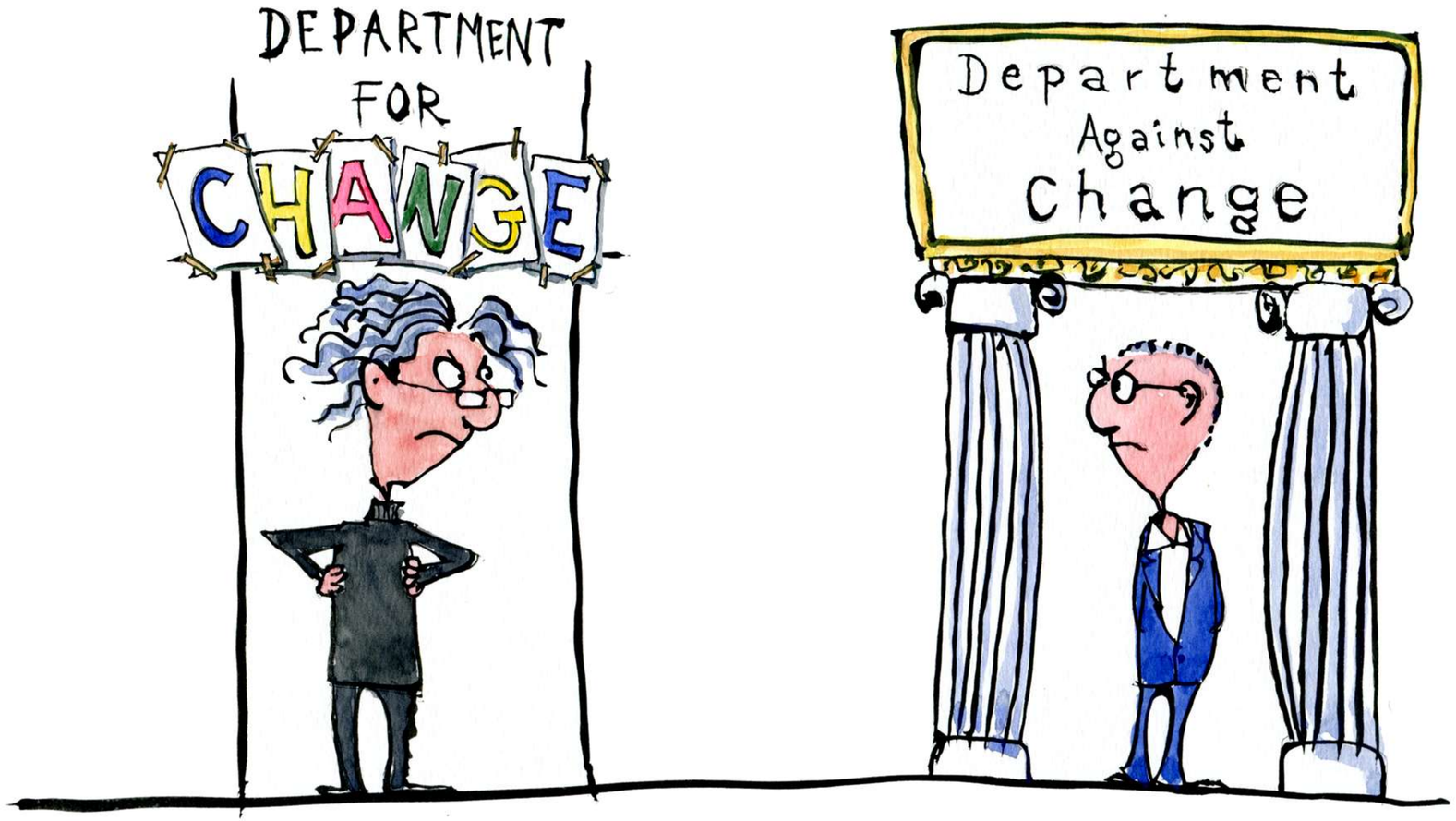


- Culture
- Automation
- Lean
- Measurement
- Sharing

Continuous Delivery
+ Operability
= DevOps


Maintainability
Resilience
Disaster Recovery
Scalability
OPERABILITY
Supportability
Deployability
Monitoring
Alerting

Why did DevOps happen?



by HikingArtist

Why did DevOps happen?

We tried to answer this question:

"How do we keep up with the demand for new features and new technologies while maintaining stability and high performance?"

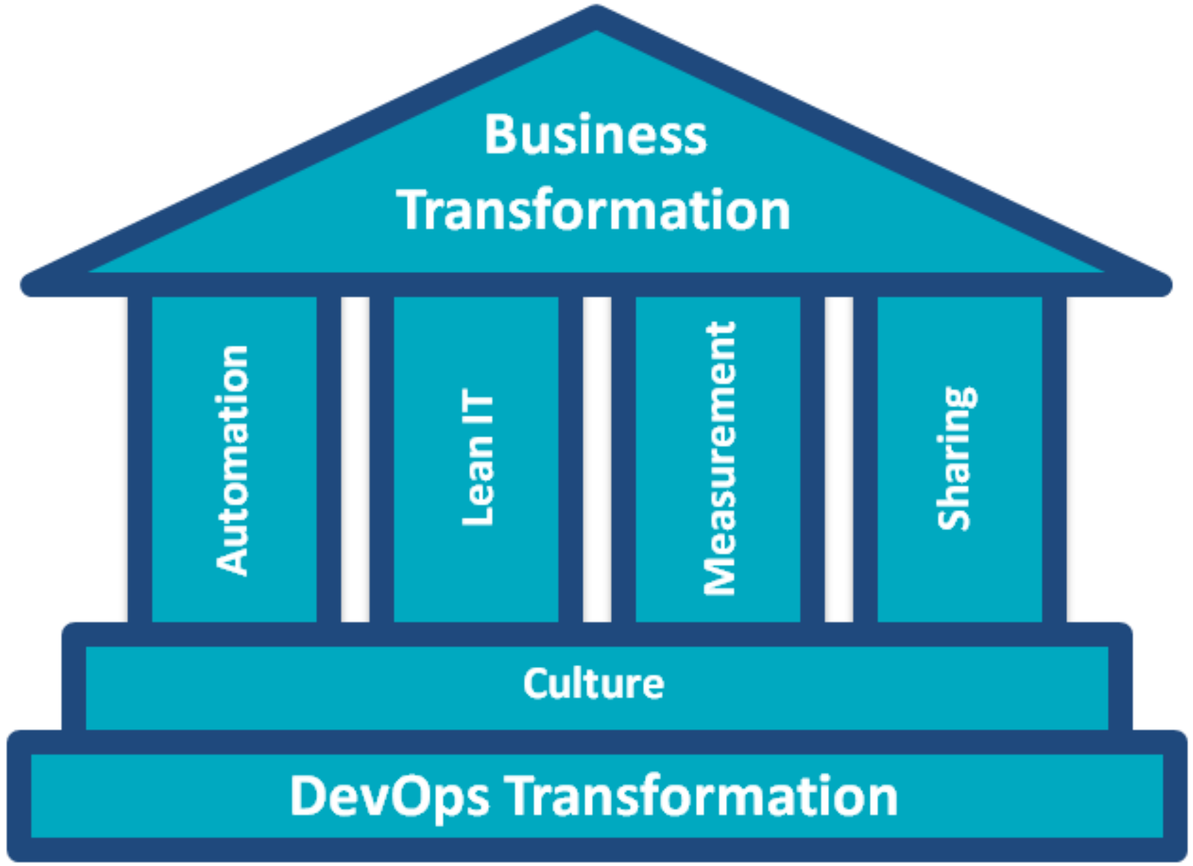
It was the wrong question ☹

"How do we deliver maximum value to our customers and shareholders?"

Devops Principles

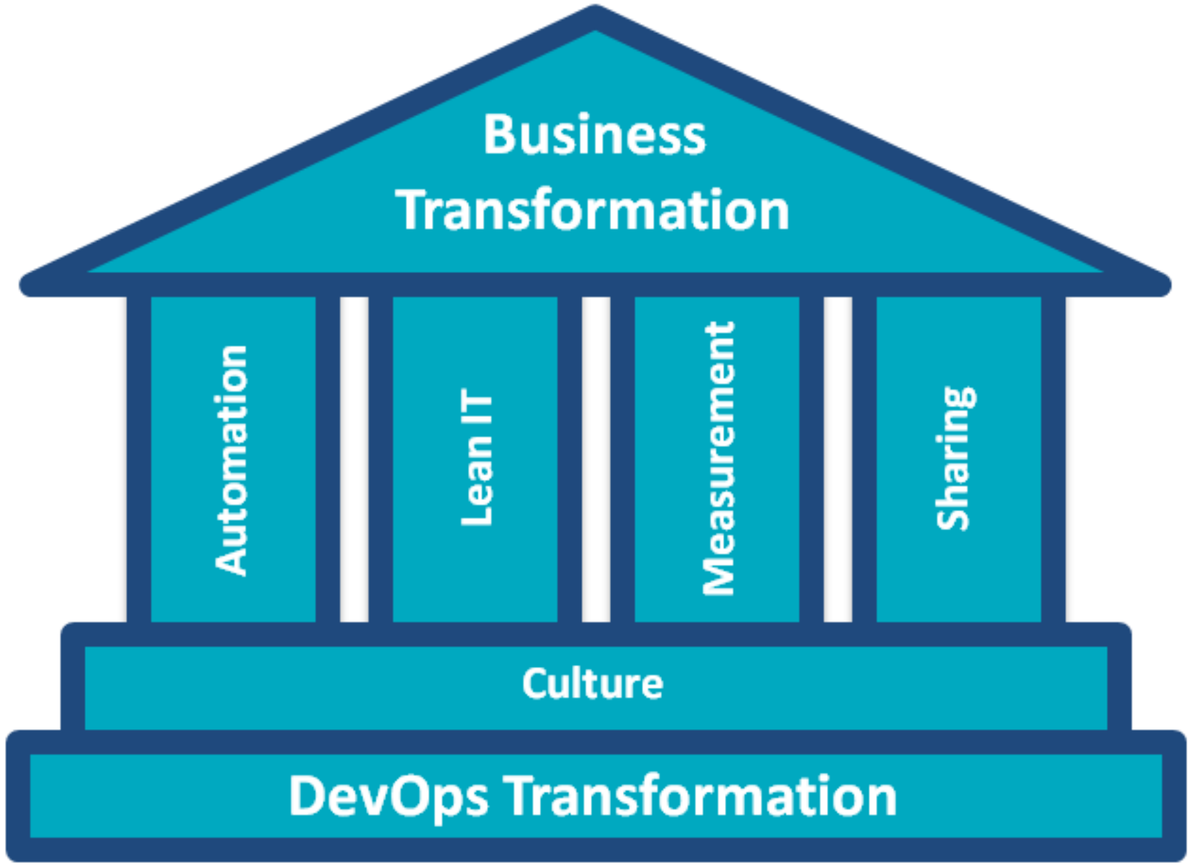
The first ingredient of DevOps:
Shared Goals

The CALMS Model



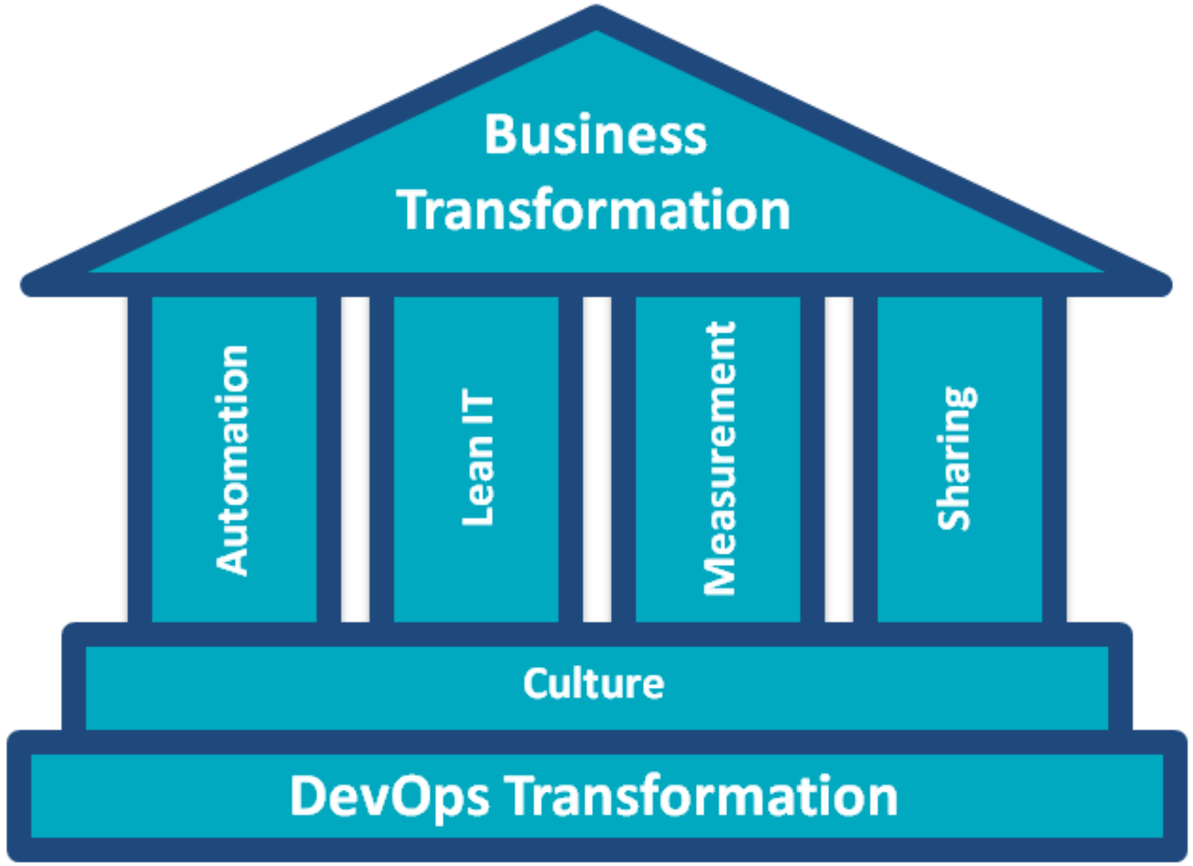
- Culture
- Automation
- Lean
- Measurement
- Sharing

The CALMS Model



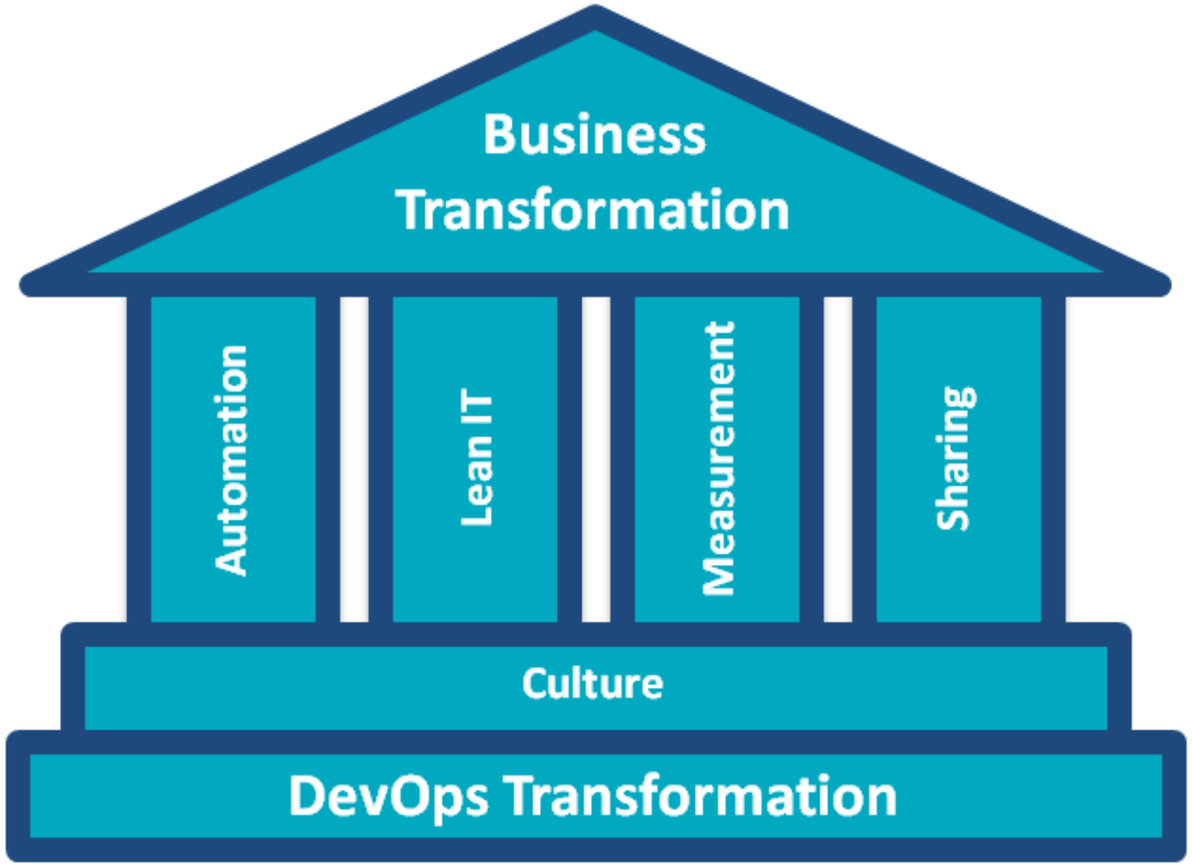
- Create a culture of collaboration and ownership.
- Start small and scale out, not up.

The CALMS Model



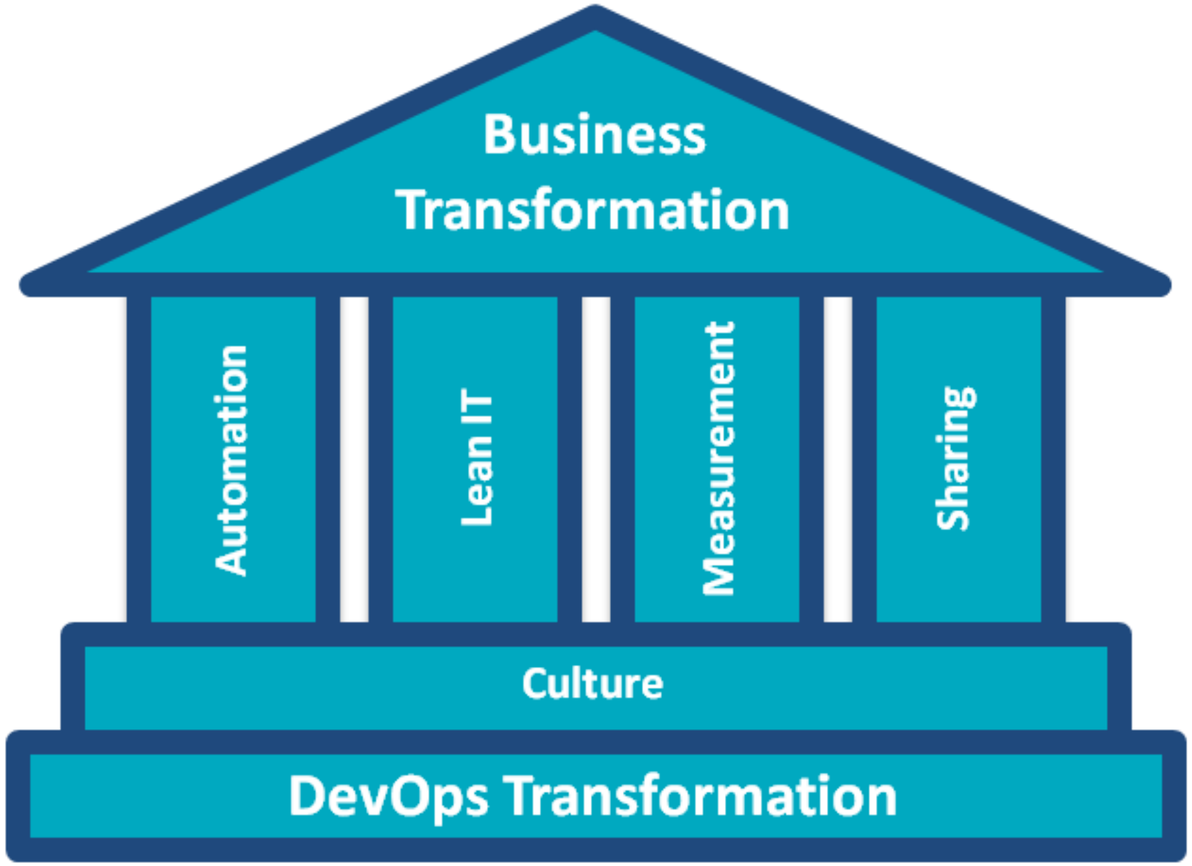
- Fast feedback through automation.
- We need information to guide our decisions

The CALMS Model



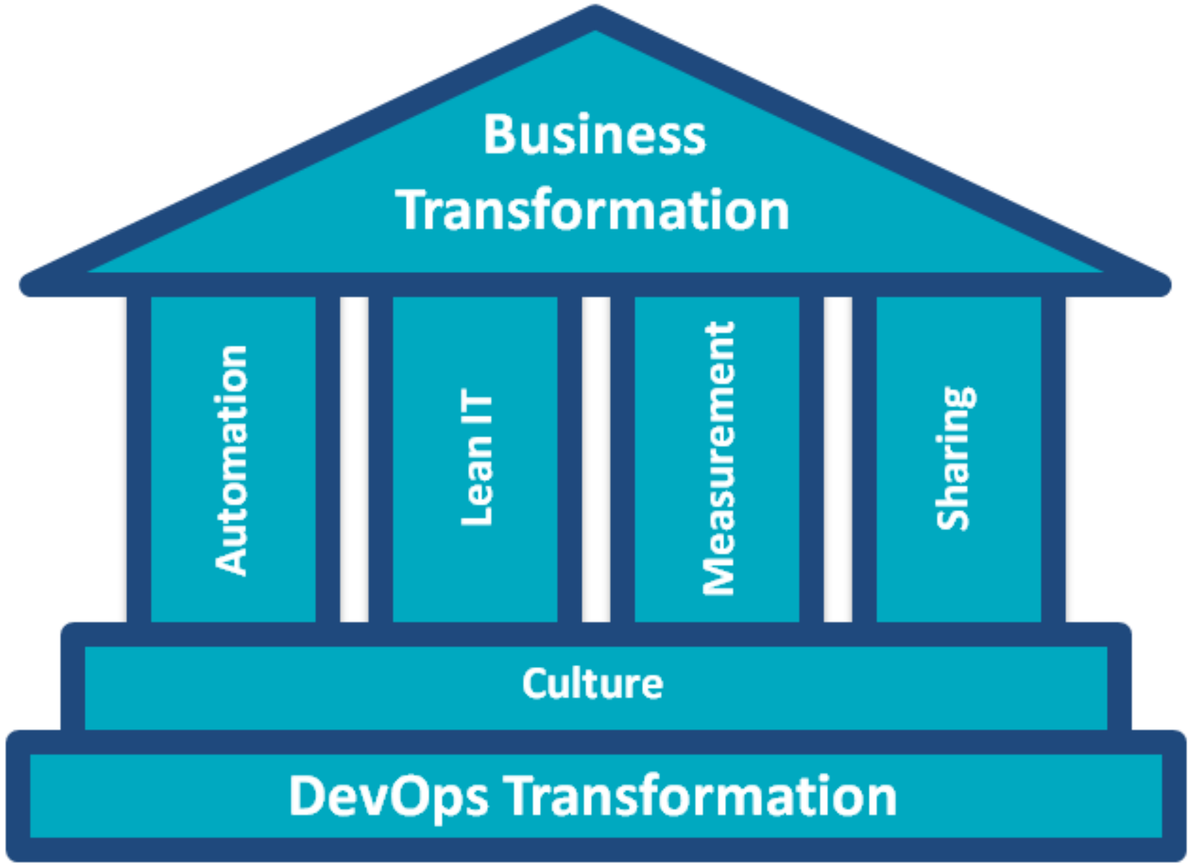
- Lean approach to system's thinking
- Localised optimisations are a mirage
- Being busy != being valuable

The CALMS Model



- Measure the right things
- Be empirical, let the stats guide you.
- Beware of the cultural impact

The CALMS Model



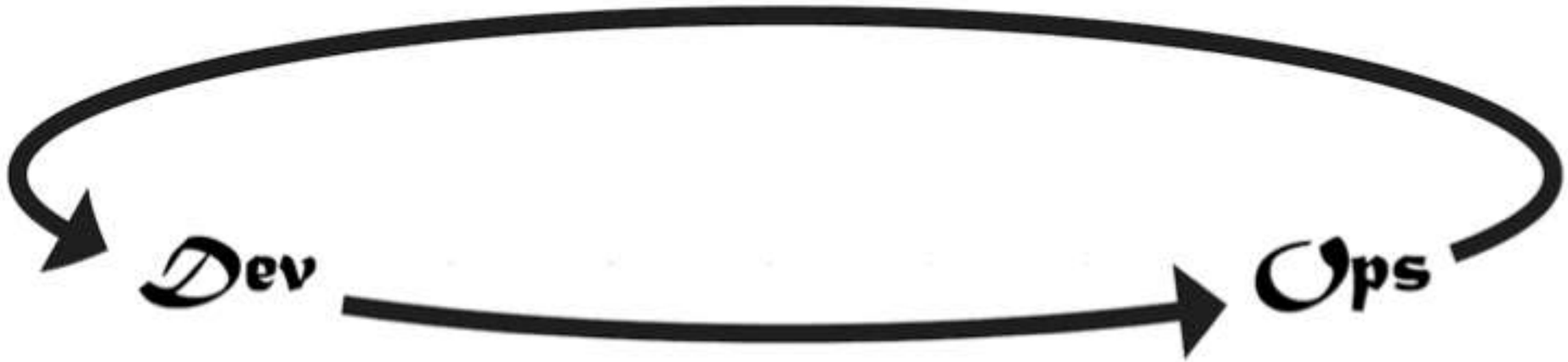
- Share goals to create a common purpose
- Share experiences to encourage learning

Gene Kim's "3 Ways" of DevOps

The First Way:
Systems Thinking

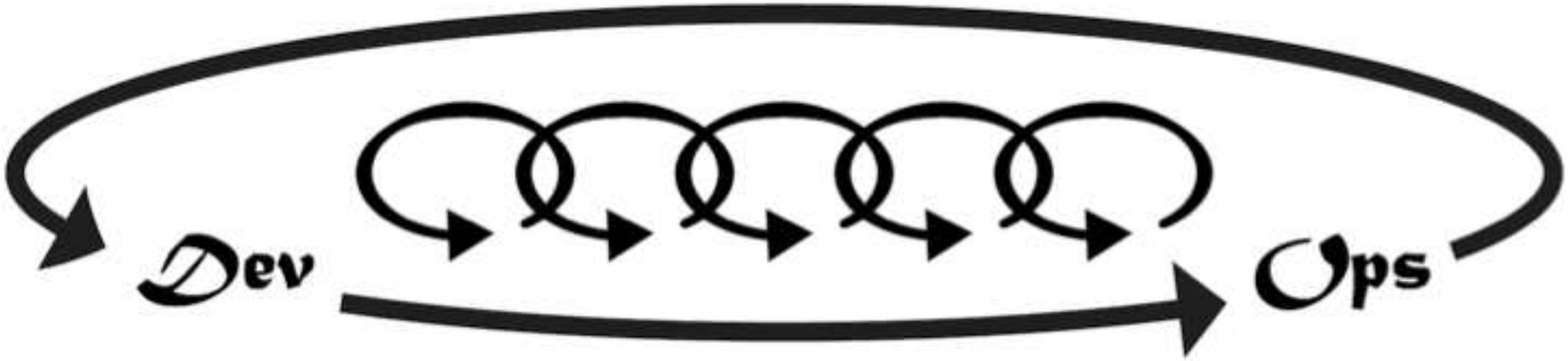
Gene Kim's "3 Ways" of DevOps

The Second Way:
Amplify Feedback Loops



Gene Kim's "3 Ways" of DevOps

The Third Way:
Culture Of Continual Experimentation And Learning



Devops Practices

Infrastructure as Code

Configuration as Code

Cloud Native / Cloud First

Test Driven

Continuous Delivery

Microservices

Infrastructure as Code

```
workflow CreateWebVM
{
    InlineScript {
        "Creating VM $($Using:VMName)"
        Select-AzureSubscription $Using:SubscriptionName
        $VM = New-AzureVMConfig -Name $Using:VMName `
                                -InstanceSize "ExtraSmall" `
                                -ImageName $Using:imageName `
                                -
AvailabilitySetName $Using:availgroup
        $VMConfig = Add-AzureProvisioningConfig -Linux `
                                                -VM $VM `
                                                -
                            -LinuxUser $Using:username
                            -SSHKeyPairs $Using:sshkey
                            -password $Using:password
 New-AzureVM -ServiceName $Using:CloudService.ServiceName -VM $VMConfig
    }
}
```

- ✓Declarative
- ✓Reusable
- ✓Automated
- ✓Testable

Configuration as Code

```
package "apache2" do
  case node[:platform]
  when "centos","redhat","fedora","suse"
    package_name "httpd"
  when "debian","ubuntu"
    package_name "apache2"
  when "arch"
    package_name "apache"
  end
  action :install
end
```

Test Driven

As a lazy ops guy I want an Ansible role that will install Apache on an Ubuntu Server So that I can host the best website ever

Additional notes:

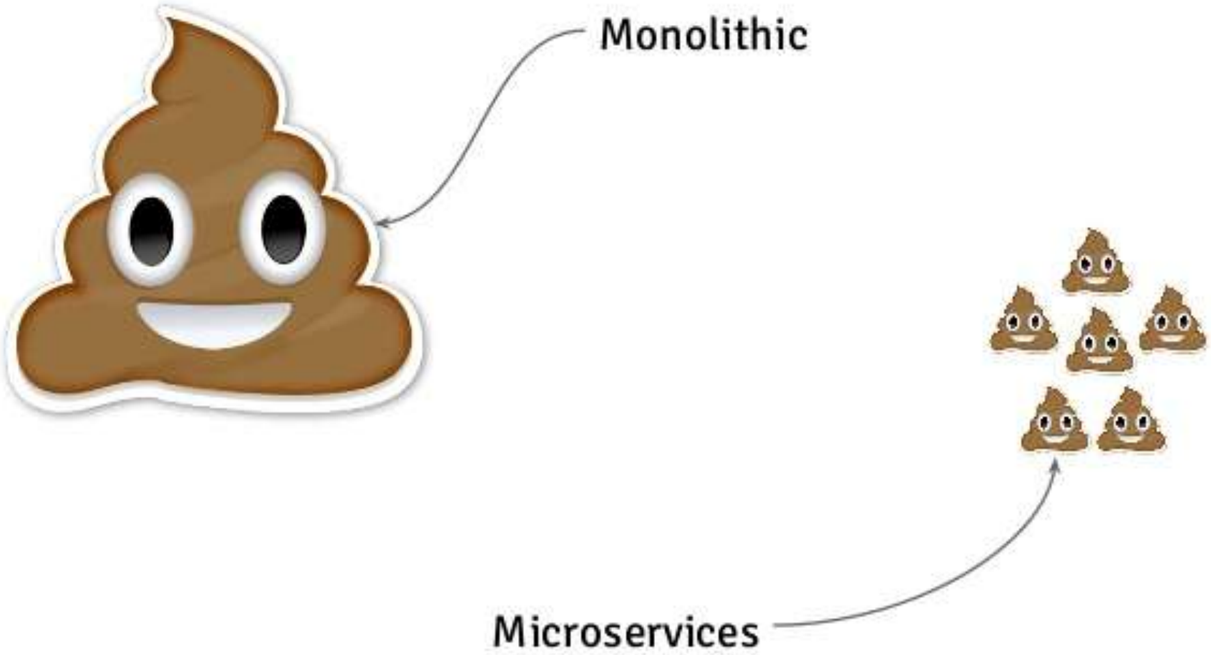
- Install whatever the latest version of Apache is, we're a bleeding edge company
- Must work on Ubuntu 14.04 (current) and 15.04 (future rollout)
- The external load balancer will route incoming http requests to port 55555 on all machines
- The NSA wants their own root account on all our environments
- Please remove telnet for maximum security

Test Driven

```
require 'spec_helper'
    describe package('apache2') do
      it { should be_installed }
    end
    describe service('apache2') do
      it { should be_running }
    end
    describe port(55555) do
    it { should be_listening }
    end
    describe user('NSA') do
      it { should exist }
      it { should belong_to_group('root') }
end
```

Microservices

Monolithic vs Microservices

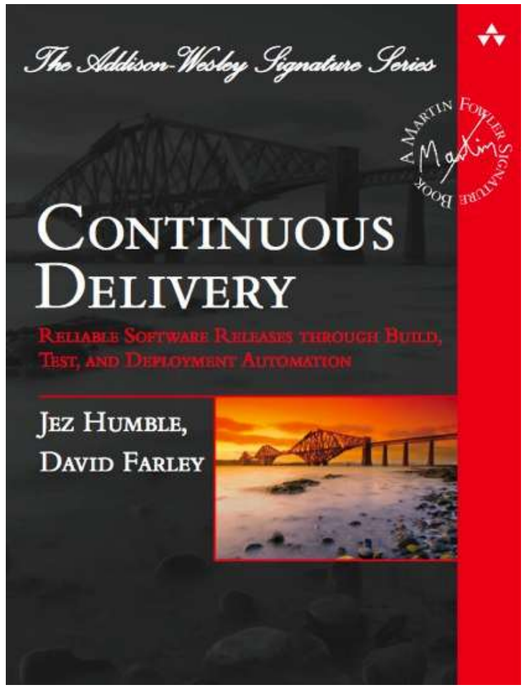


@alvaro_sanchez

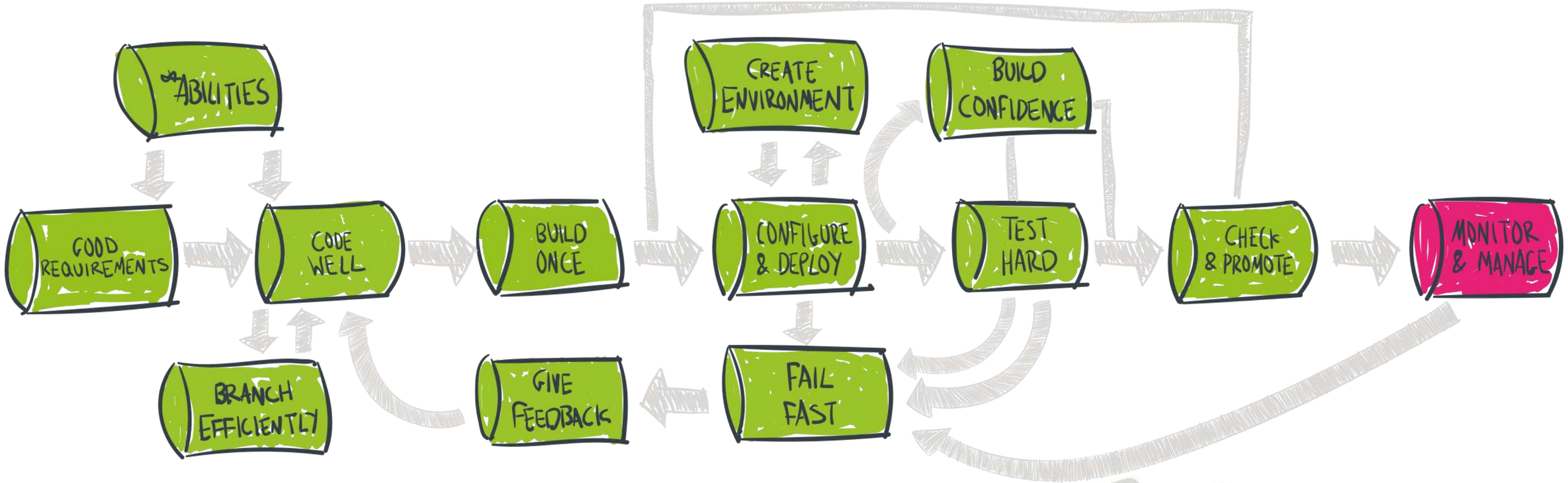
odobo

Continuous Delivery

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



Continuous Delivery



Continuous Delivery – 8 Principles

- The process for releasing/deploying software MUST be repeatable and reliable
- Automate everything!
- If something is difficult or painful, do it more often
- Keep everything in source control
- Done means “released”
- Build quality in!
- Everybody has responsibility for the release process
- Improve continuously

Continuous Delivery – 4 Practices

- **Build Binaries only once**
- **Use precisely the same mechanism to deploy to every environment**
- **Smoke test your deployments**
- **If anything fails, stop the process.**

PERIODIC TABLE OF DEVOPS TOOLS (V1)

XebiaLabs
Deliver Faster

Os	Open Source
Fr	Free
Fm	Freemium
Pd	Paid
En	Enterprise

Database	SCM	Build
CI	Repo Mgmt	Testing
Deployment	Config / Provisioning	Containerization
Cloud / Iaas / Paas	Release Mgmt	Collaboration
BI / Monitoring	Logging	Security

1 En O 12c																		2 Fm Aws Amazon Web Services
3 Os My MySQL	4 Os Gt Git												5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 En Sl Salt	9 Os Dk Docker	10 Pd Az Azure
11 En Mq MSSQL	12 Os Sv Subversion												13 Fr Ssh SSH	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk rkt	18 Fm Hk Heroku
19 Os Pq PostgreSQL	20 Fr Mc Mercurial	21 Os Mv Maven	22 Os Gr Gradle	23 En Mr Meister	24 Os Jn Jenkins	25 Pd Bb Bamboo	26 Os Tr Travis CI	27 Fr Ar Archiva	28 Os Fn FitNesse	29 Fr Se Selenium	30 Os Gn Gatling	31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Fr Cb Cobbler	34 Os Bc Bcfg2	35 Os Kb Kubernetes	36 En Rs Rackspace
37 Os Mg MongoDB	38 Fm Gh Github	39 Os Br Buildr	40 Os At ANT	41 Fm Bm BuildMaster	42 Fm Cs Codeship	43 Fm Sn Snap CI	44 Fm Cr CircleCI	45 Os Nx Nexus	46 Fr Cu Cucumber	47 Os Cj Cucumber.js	48 Fr Qu Qunit	49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Pk Packer	54 Fm Bx Bluemix
55 En Db DB2	56 Fm Bb Bitbucket	57 Fm Qb QuickBuild	58 En Ub UrbanCode Build	59 Pd Ta Visual Build	60 Fm Tc TeamCity	61 Fm Sh Shippable	62 Os Cc CruiseControl	63 Os Ay Artifactory	64 Fr Ju JUnit	65 Fr Jm JMeter	66 Fr Tn TestNG	67 En Rd RapidDeploy	68 Fm Cy CodeDeploy	69 En Oc Octopus Deploy	70 No CA Nolio	71 En Eb ElasticBox	72 En Ad Apprenda
73 Fr Cs Cassandra	74 En Hx Helix	75 Os Msb MSBuild	76 Os Rk Rake	77 Os Lb LuntBuild	78 Os Cu Continuum	79 Fm Ca Continua CI	80 Os Gu Gump	81 Os Ng NuGet	82 Os Ap Appium	83 En Xltv XL TestView	84 En Tc TestComplete	85 Os Go Go	86 En Ef ElectricFlow	87 En Xld XL Deploy	88 En Ud UrbanCode Deploy	89 Os Mo Mesos	90 Os Cf Cloud Foundry

Share

Embed

Become Excellent!

Subscribe here!

91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Ls CA Service Virtualization	94 En Bm BMC Release Process	95 En Hp HP Codar	96 Pd Ex Excel	97 En Pl Plutora Release	98 En Sr Serena Release	99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow
106 En Sp Splunk	107 Os Ki Kibana	108 Fm Nr New Relic	109 Os Ni Nagios	110 Os Gg Ganglia	111 Os Ct Cacti	112 Os Gr Graphite	113 Os Ic Icinga	114 Fm Sl Sumo Logic	115 Os Ls Logstash	116 Fm Lg Loggly	117 Os Gr Graylog	118 Os Sn Snort	119 Os Tr Tripwire	120 En Cy CyberArk

How to do Devops in your organization

What do these have in common?

5 Steps to do Devops

- Establish your goals
  - What does DevOps mean to the team?

- Build the platform
  - Environments
  - Continuous Delivery
  - Test Automation

- Assemble the team

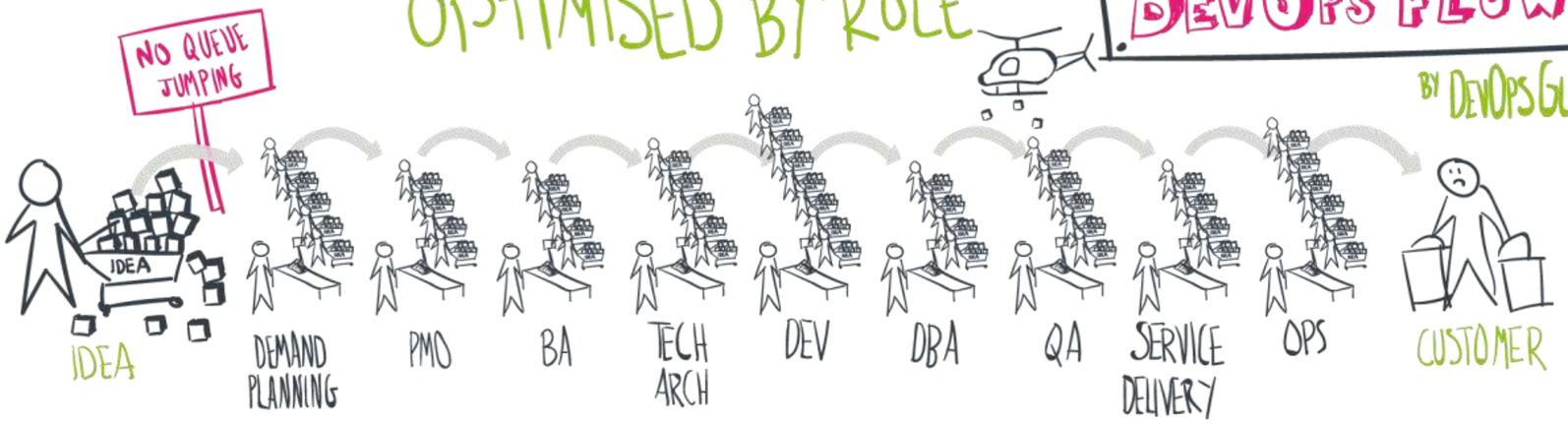
- Be agile, not waterfall

- Work together to achieve great things
  - Autonomy, mastery & purpose

DevOps Flow

by DevOps Guys

Optimised by Role

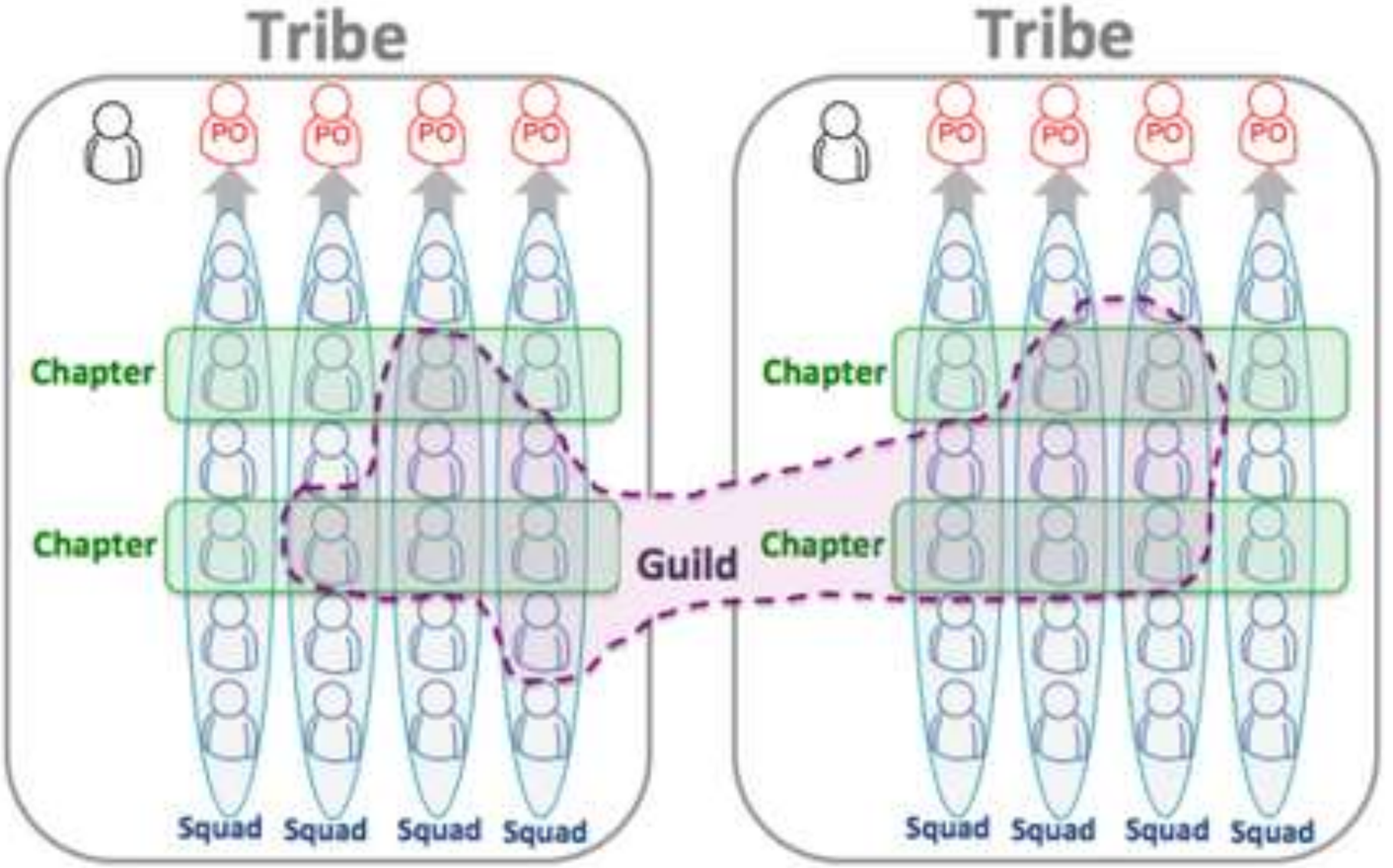


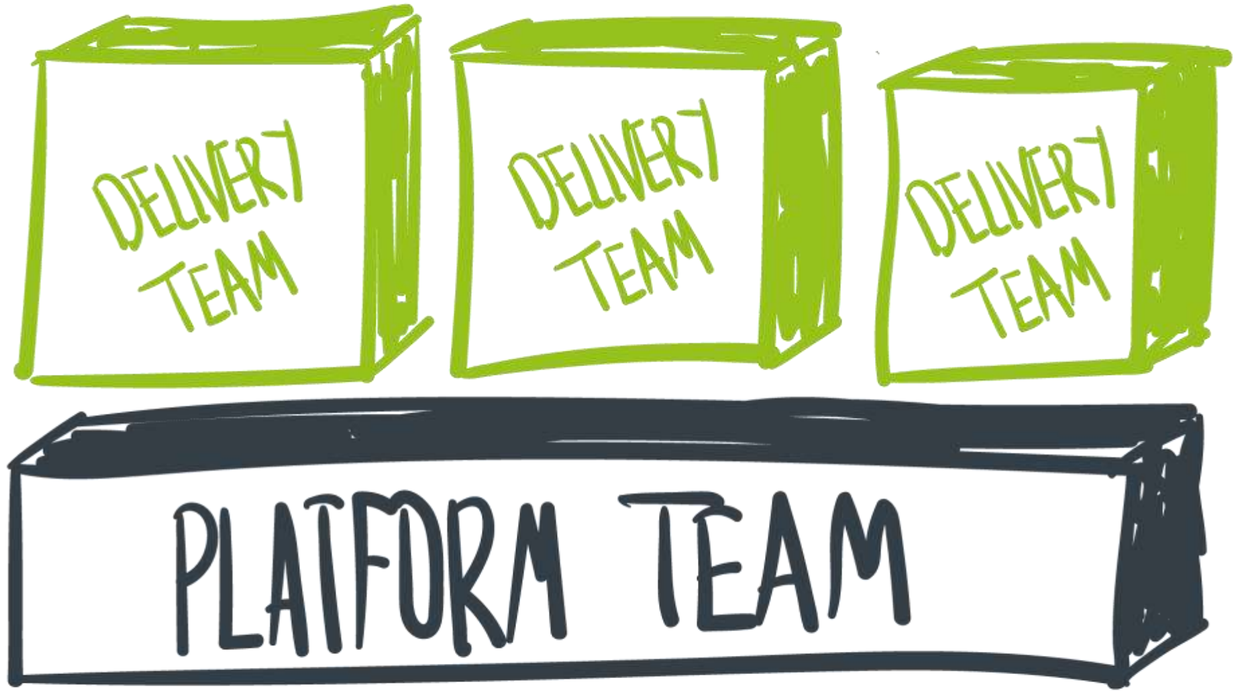
Optimised for Flow

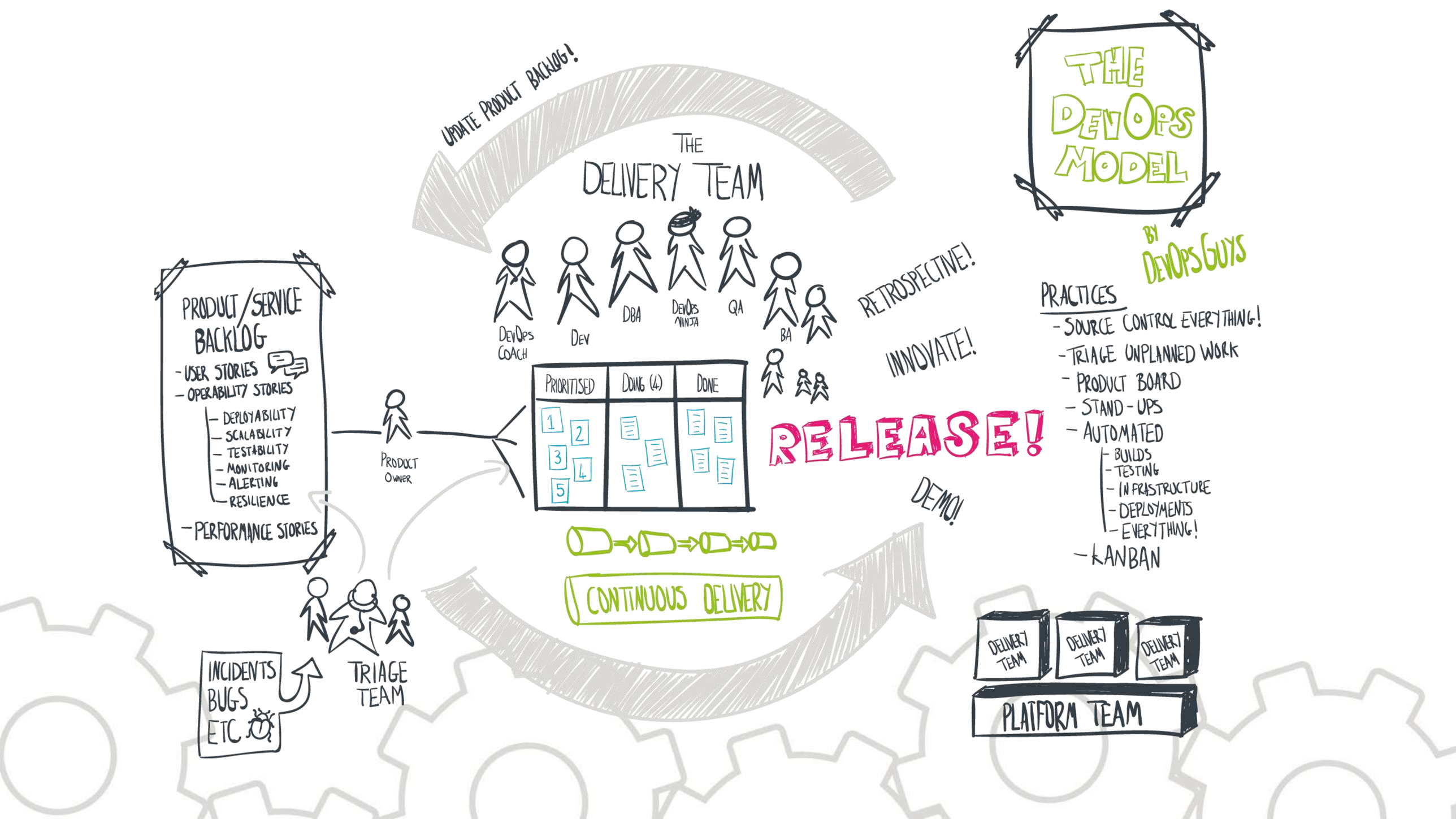
- Reduce Handovers
- Remove Queues
- Lower Batch Size
- Improve Cycle Time

DevOpsGuys

Henrik Kniberg & Anders Ivarsson
Oct 2012




DELIVERY TEAM
DELIVERY TEAM
DELIVERY TEAM
PLATFORM TEAM


THE DEVOPS MODEL
BY DEVOPS GUYS
UPDATE PRODUCT BACKLOG!
THE DELIVERY TEAM
DEVOPS COACH
DEV
DBA
DEVOPS NINJA
QA
BA
RETROSPECTIVE!
INNOVATE!
PRODUCT/SERVICE BACKLOG
- USER STORIES
- OPERABILITY STORIES
- DEPLOYABILITY
- SCALABILITY
- TESTABILITY
- MONITORING
- ALERTING
- RESILIENCE
- PERFORMANCE STORIES
PRODUCT OWNER
PRIORITISED
DOING (4)
DONE
1
2
3
4
5
RELEASE!
DEMO!
CONTINUOUS DELIVERY
PRACTICES
- SOURCE CONTROL EVERYTHING!
- TRIAGE UNPLANNED WORK
- PRODUCT BOARD
- STAND-UPS
- AUTOMATED
- BUILDS
- TESTING
- INFRASTRUCTURE
- DEPLOYMENTS
- EVERYTHING!
- KANBAN
INCIDENTS BUGS ETC
TRIAGE TEAM
DELIVERY TEAM
DELIVERY TEAM
DELIVERY TEAM
PLATFORM TEAM

The Devops Solution

DevOps Coaching

Workshops & Training

DevOps Engineering

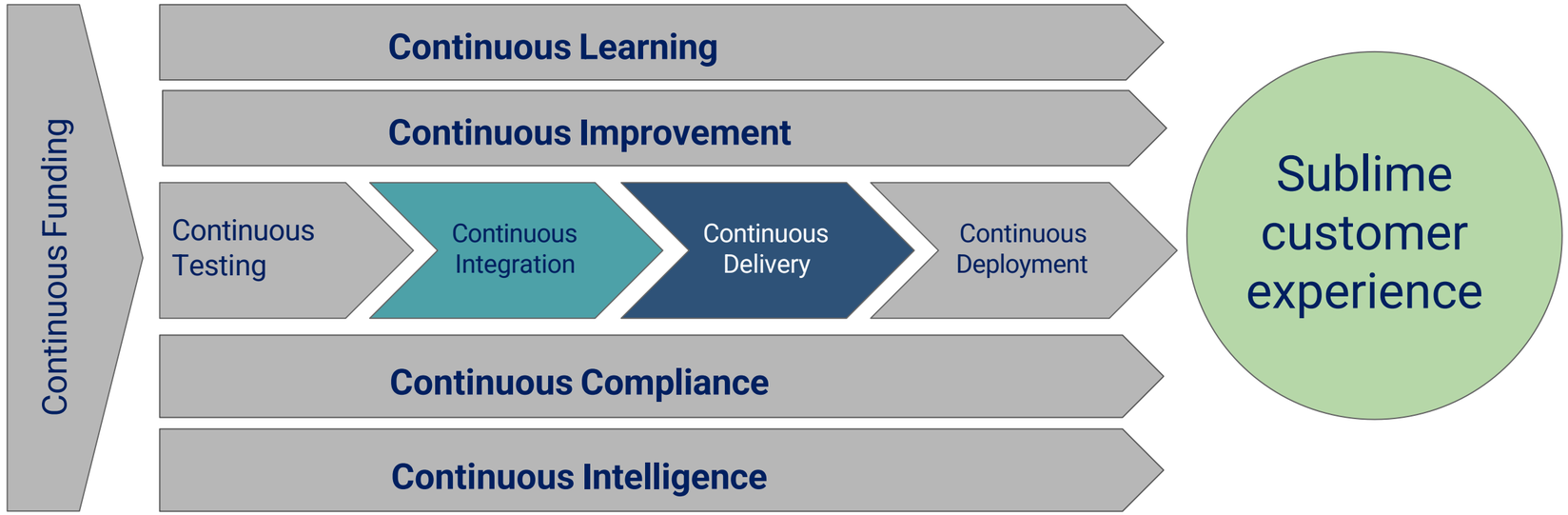
Application Lifecyle Automation

DevOps Consultancy

DevOps, Agile & Cloud Strategy

CI and CD

Devops Practices – All the continuouses



Core Must Have Automation Skills

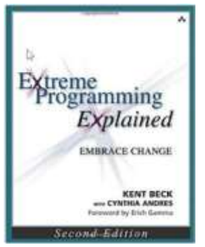
CI + CD are leading valuable skills

Continuous Integration Defined

Continuous Integration is a software development practice where members of a team integrate their work frequently.

In most instances, each person of the team integrates their code at least daily - leading to multiple integrations per day.

Each integration is verified by an automated build and test in order to detect integration errors as quickly as possible.



Kent Beck
1999

Paul Duvall

Continuous Integration Defined

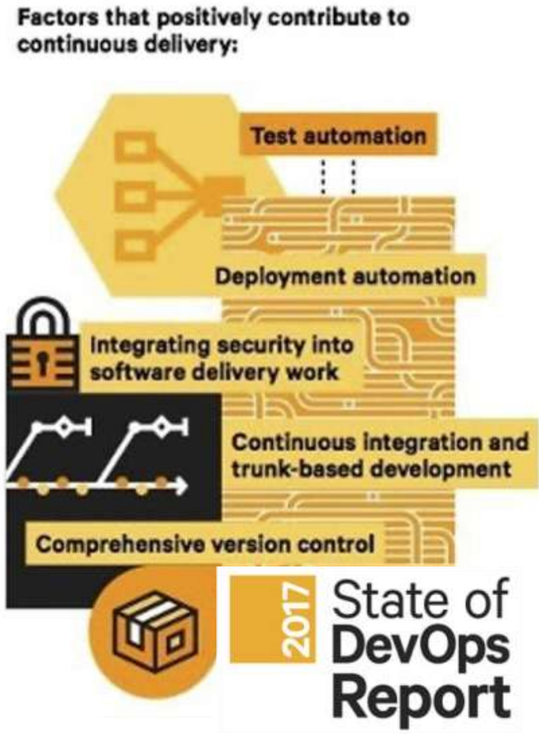
Avoid ‘merge hell’

- *All developers check code in at least daily to trunk*
- *Trunk based development*
- *Each check-in is validated by*
  - *An automated build*
  - *Automated unit, integration and acceptance tests*
- *Is Dependent on consistent coding standards.*
- *Requires version control repositories and CI servers to collect, build and test committed code together*
- *Runs on production-like environments*
- *Allows for early detection and quick remediation of errors from code changes before moving to production*

You can do this in waterfall too… if you want to

Continuous Delivery

- *Takes continuous integration to the next level*
- *Provides fast, automated feedback on a system's production-readiness*
- *Prioritizes keeping software releasable/deployable over working on new features*
- *Relies on a deployment pipeline that enables push-button deployments on demand*
- *Reduces the cost, time, and risk of delivering incremental changes*



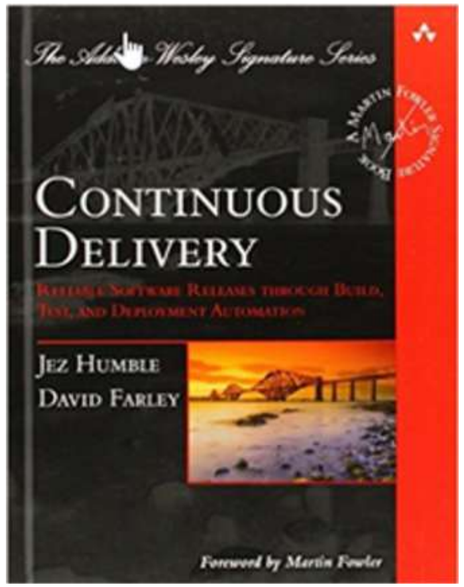
Software is always in a releasable state – just push the button !

Continuous Delivery

Continuous delivery (**CD**) is a software engineering approach [associated with DevOps,] in which **teams produce software in short cycles**, ensuring that the software can be **reliably released at any time**. It aims at building, testing, and releasing software **faster and more frequently**.

The approach helps **reduce the cost, time, and risk of delivering changes** by allowing for more **incremental updates to applications in production**. A **straightforward and repeatable deployment process** is important for continuous delivery.

Continuous Delivery

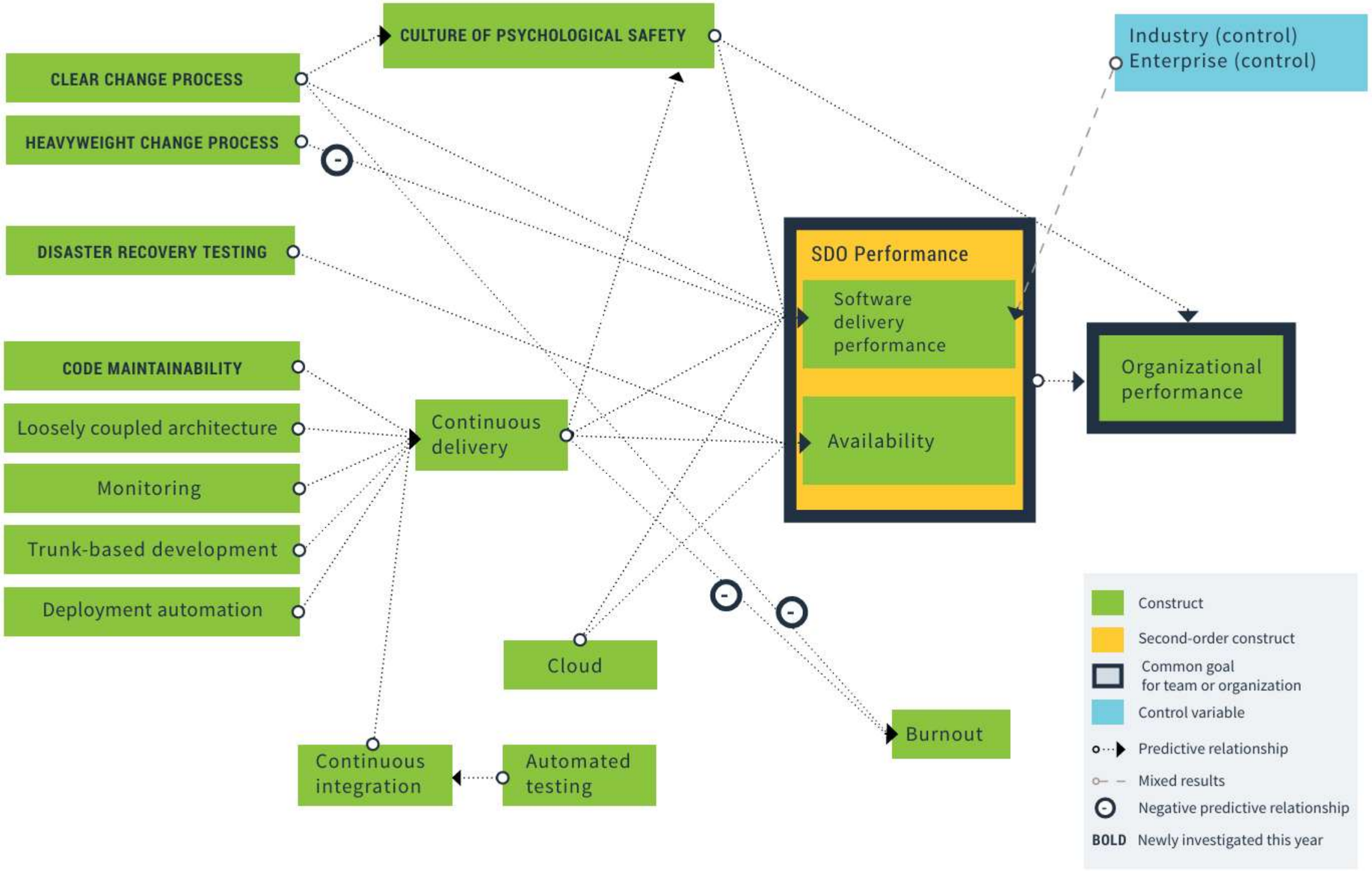


"The ability to get changes - features, configuration changes, bug fixes, experiments - **into production or into the hands of users safely and quickly in a sustainable way** "

Jez Humble
Author of "Continuous Delivery"
co-author of The DevOps handbook"

Continuous Delivery

Leads to higher organizational performance



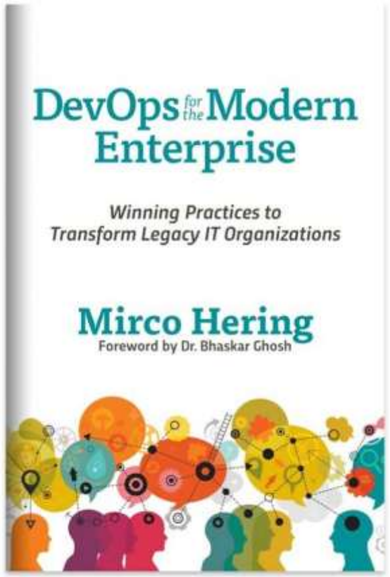
Continuous Delivery vs Continuous Deployment

Automatic trigger

Manual trigger

Continuous Deployment

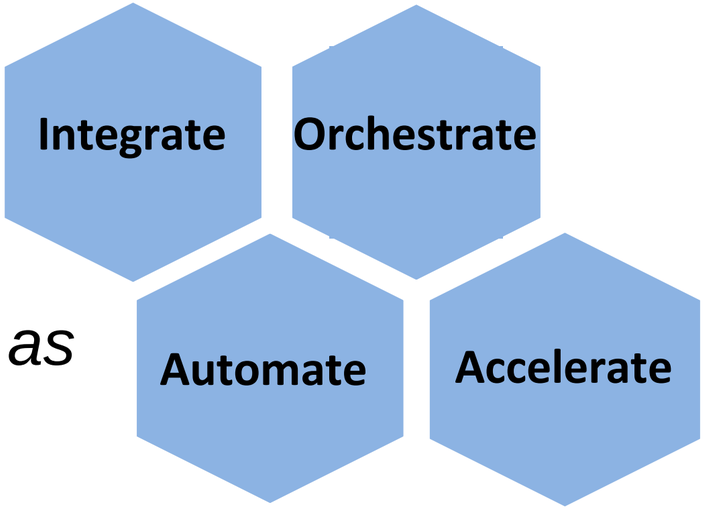
Continuous Delivery



From: Mirco Hering: notafactoryanymore.com, author of ‘DevOps for the Modern Enterprise’

Key Elements - Continuous Delivery

- *CD uses an integrated infrastructure*
- *CD emphasizes orchestration of the environment*
- *CD tasks are automated as much as possible*
- *CD goal is to accelerate activities as early in the pipeline as possible*



What do you lose?

☒ *Application quality issues*

☒ *Complex merge issues*

☒ *Security events*

☒ *Pipeline failures*

☒ *Interruptive reverts*

☒ *Process delays*

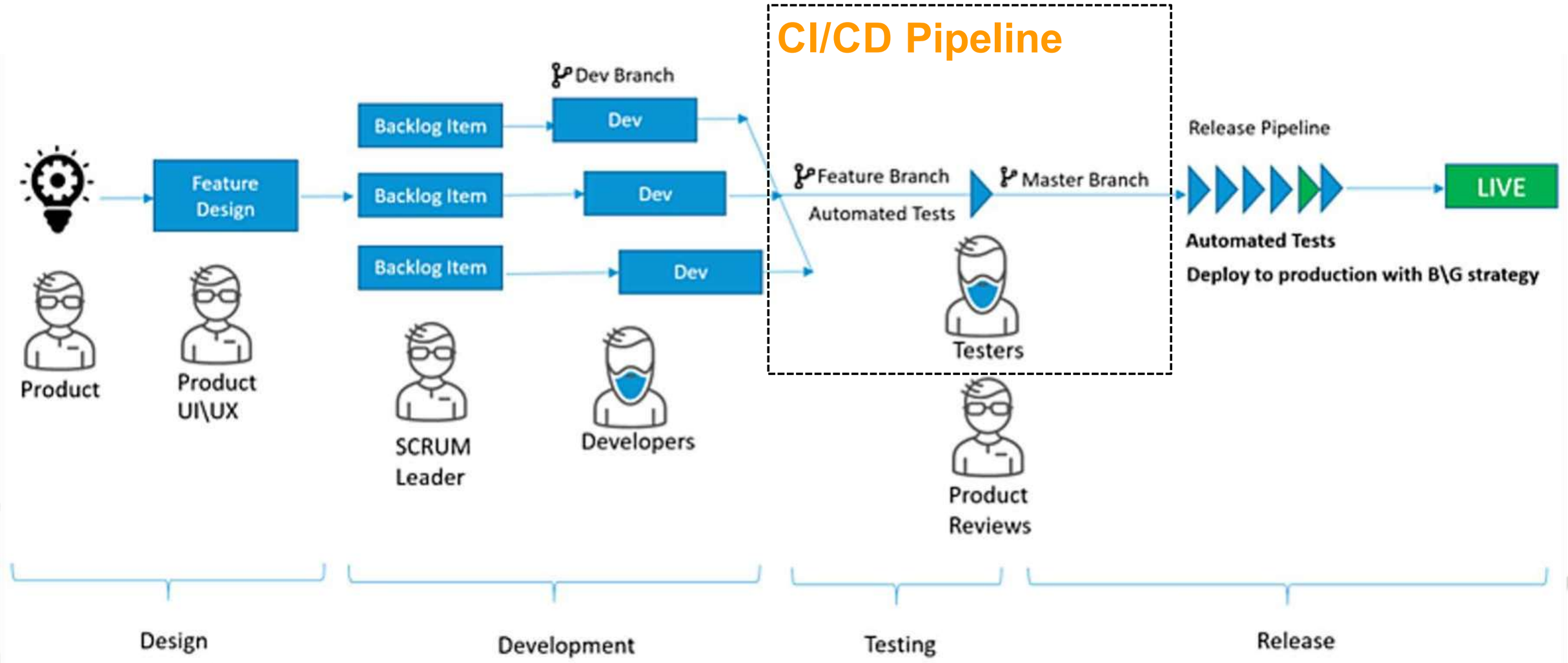
☒ *Schedule delays*

☒ *Cost overruns*

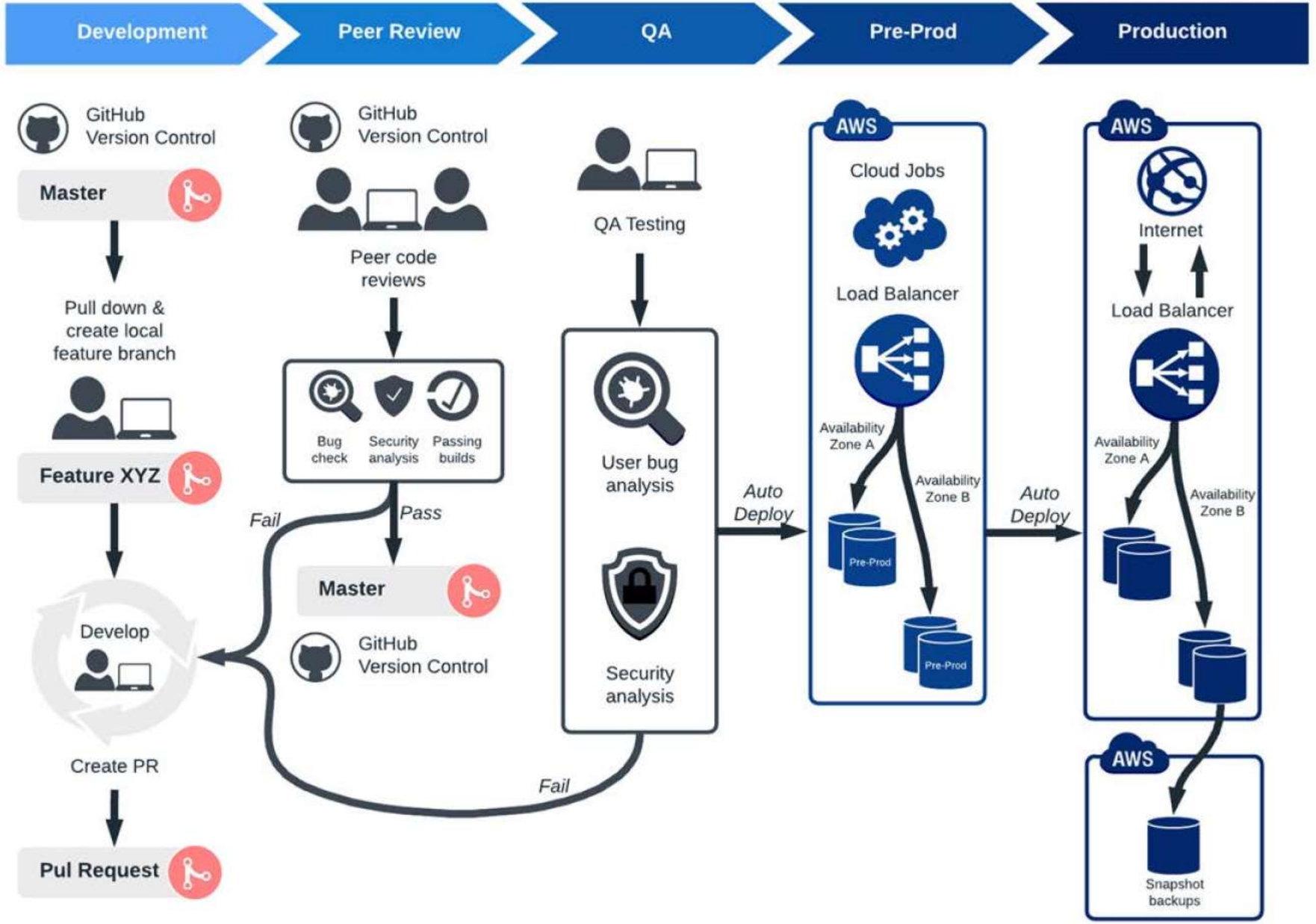
☒ *Poor morale / unhappiness*

☒ *Audit failures*

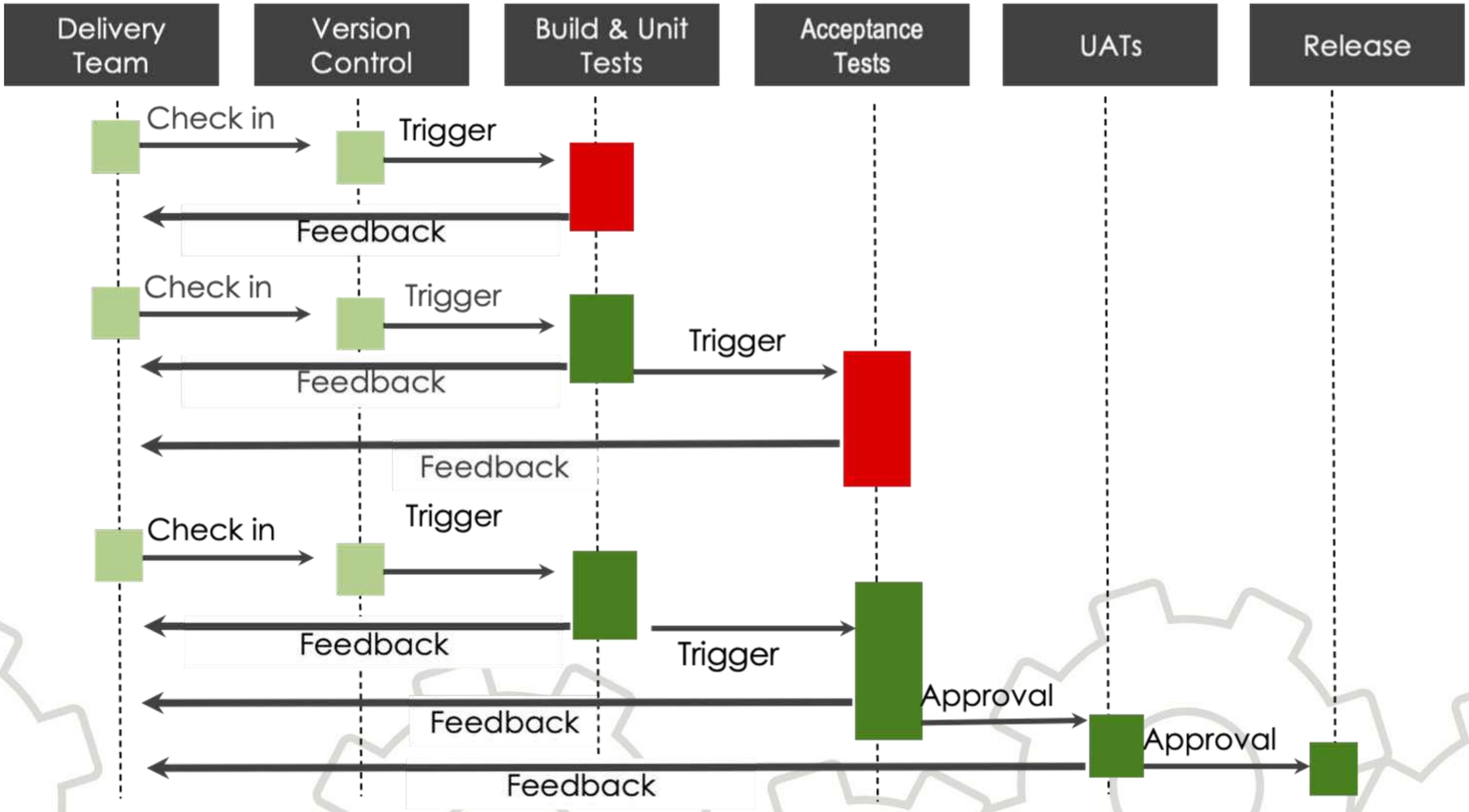
A Sample Process



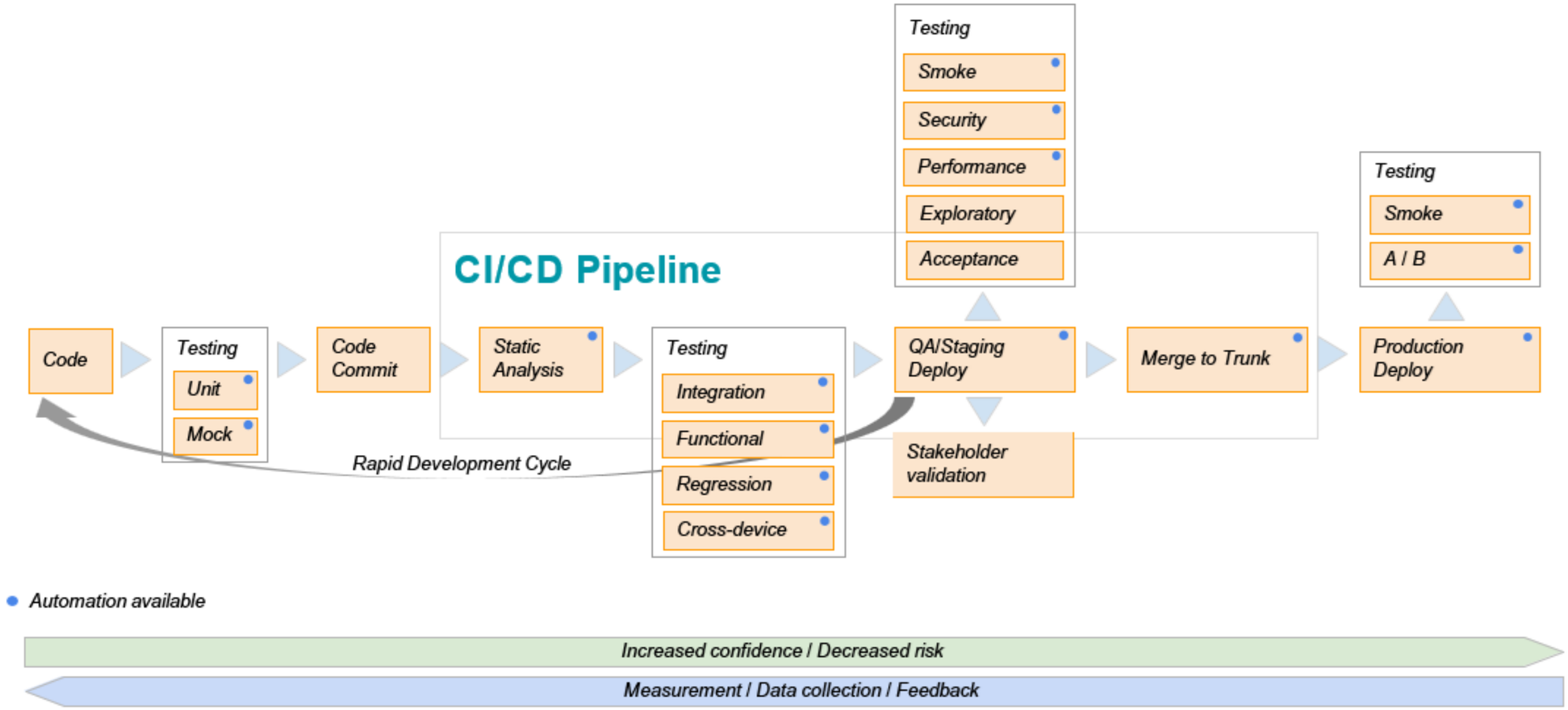
Pipeline workflow



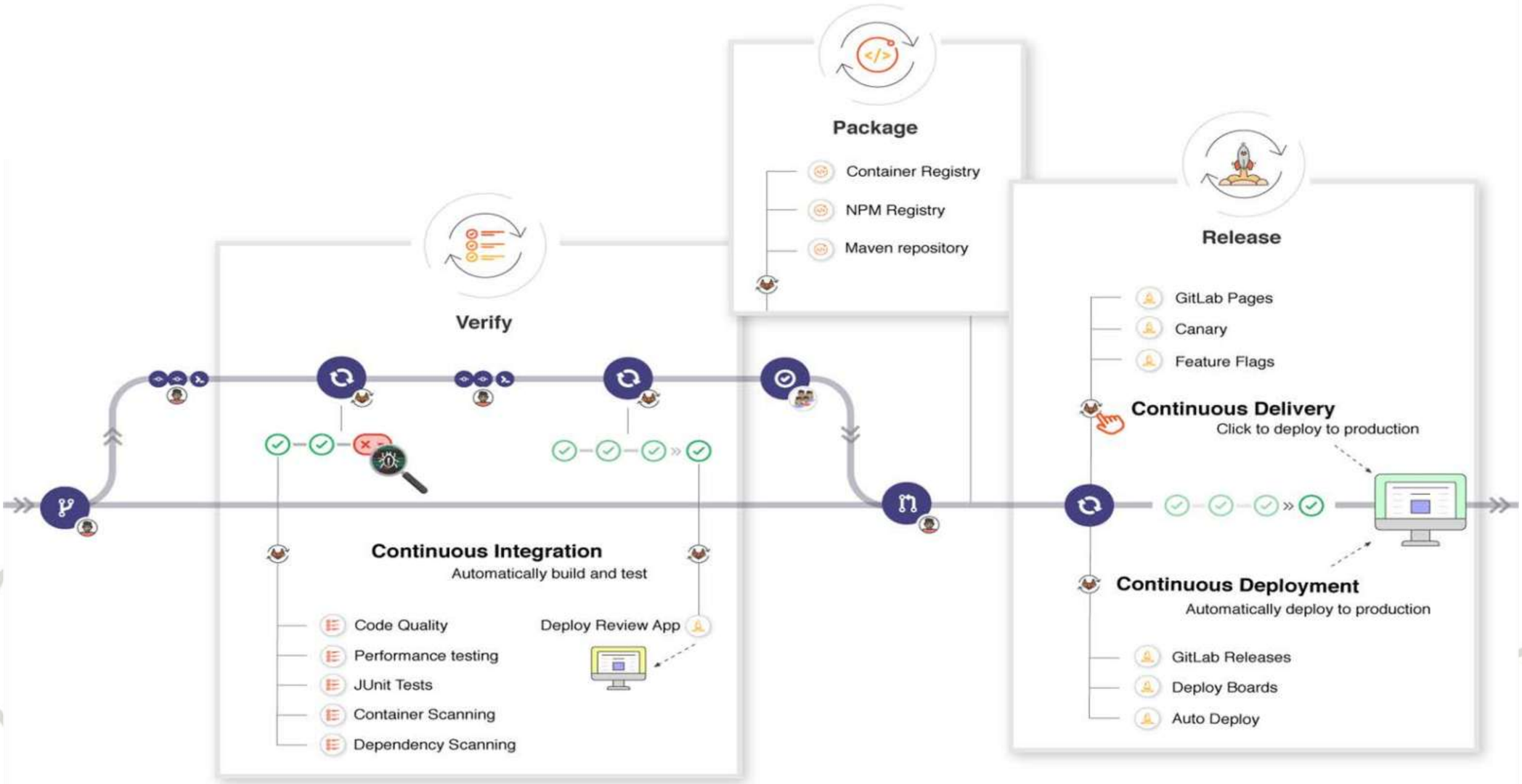
The Delivery Pipeline



Sample Pipeline



Pipeline Examples



The Goals really are

Define your pipelines to begin with.

Measure the pipeline and steps in the pipeline

Always aim at cutting down the time and the number of steps

Evaluate if any step is adding or reducing value to the process – change the process.

Pre-Flight Testing Best Practices

Before committing changes to branch,

- Run Static Analysis
- Peer Review Source Code
- Run Unit Tests
- Run Functional Tests
- Run Pre-Flight Tests in an environment that’s equivalent to prod. Environment.

Core Tools

A Version Control System

A Server to implement and Run the Pipelines

Faster Deployment of Infra via lightweight cntainers