Combining Speed of Delivery and Quality in Complex Systems

Manuel Pais | DevOps & Delivery Consultant
@manupaisable | manuelpais.net

About me

Manuel Pais

MS Software Eng Carnegie Mellon University

@manupaisable

manuelpais.net

manuel.pais@gmail.com

DevOps and Delivery Consultant

Focused on teams and flow

About me

Co-author:

Team Guide to Software Releasability

by Chris O'Dell & Manuel Pais

POCKET-SIZED INSIGHTS FOR SOFTWARE TEAMS
TEAM GUIDE TO
SOFTWARE RELEASABILITY
Chris O'Dell & Manuel Pais
4
SKELTON THATCHER
PUBLICATIONS

releasabilitybook.com

About me

DevOps Topologies

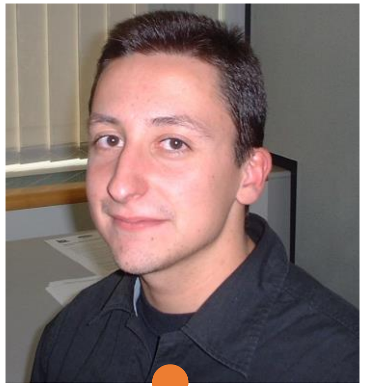
Anti-Types Team Topologies

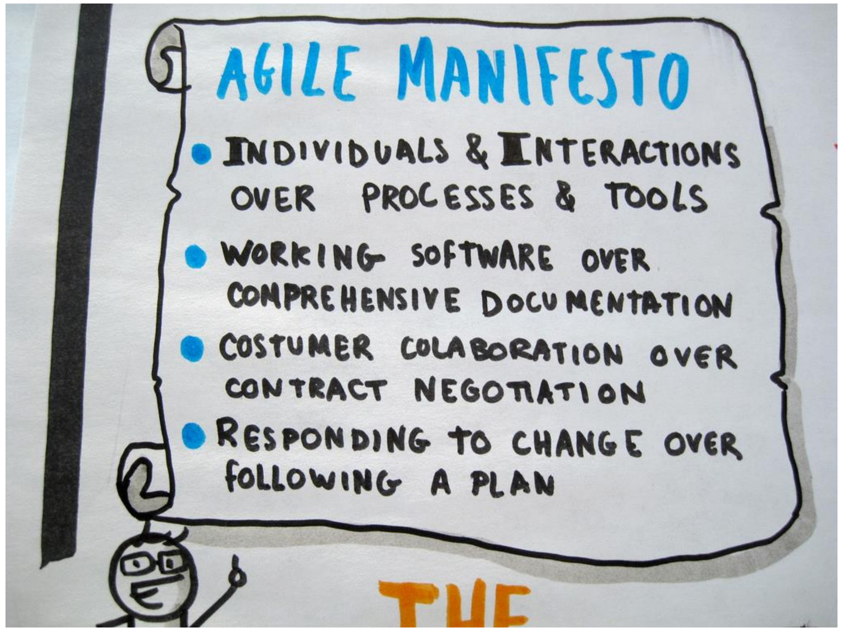
What Team Structure is Right for DevOps to Flourish?

Agenda

1. The Need for Speed (aka DevOps)
2. Failure = Quality in Complex Systems
3. Survival of High-Performing Cultures

2001: Agile Manifesto





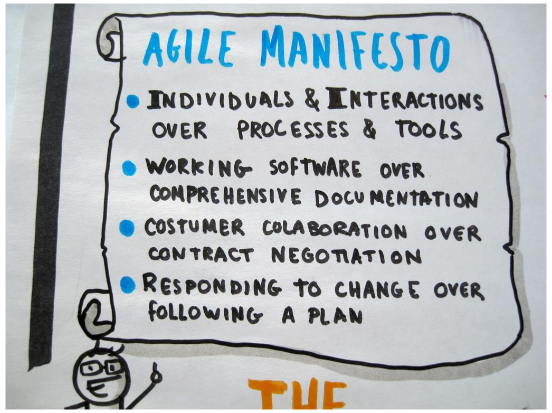
AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

THE

Java / Web Developer
2003
2008
2014
2017

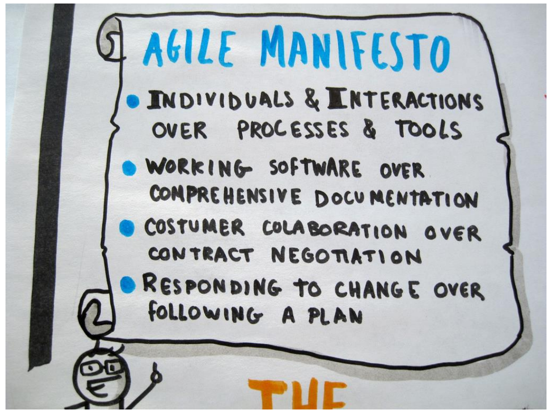
Agile... Scrum



AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

Agile... Scrum



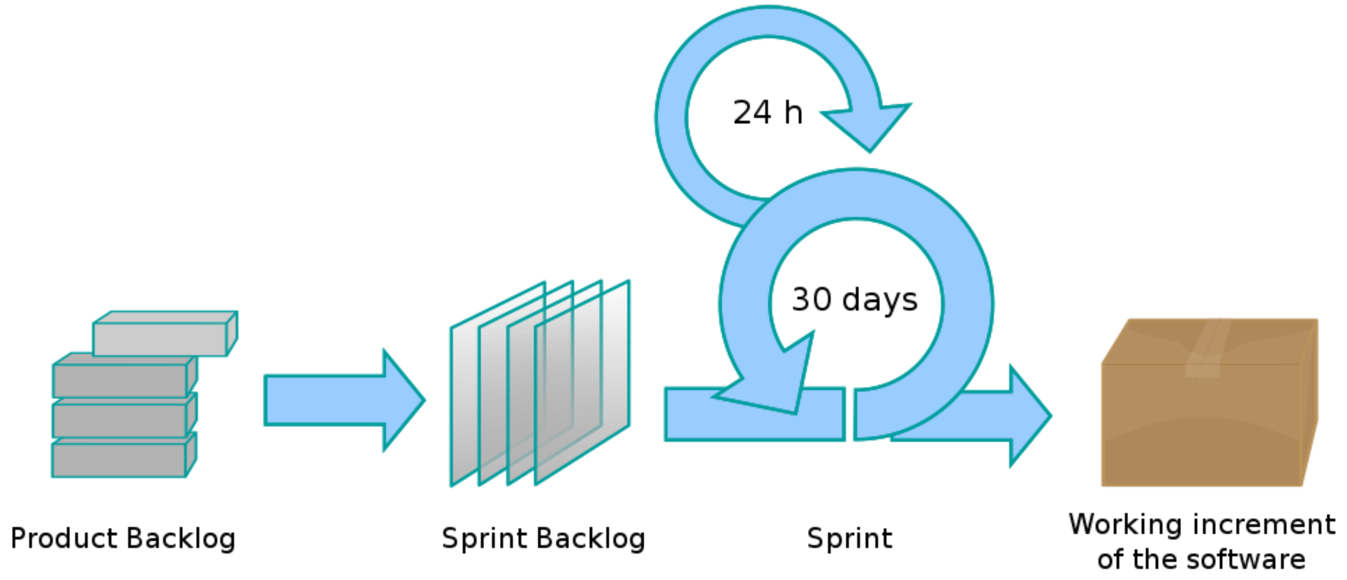
AGILE MANIFESTO

1

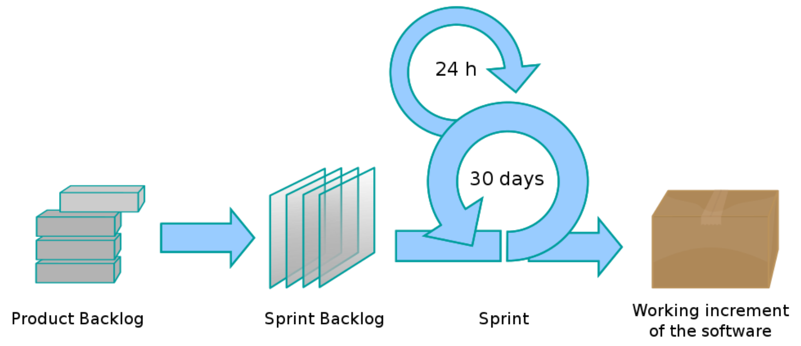
- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

2

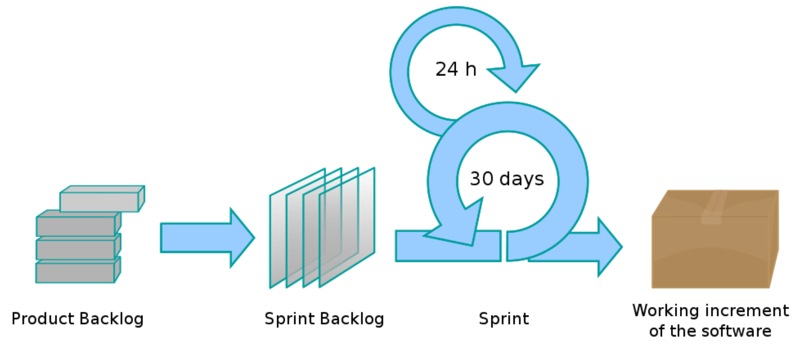
THE

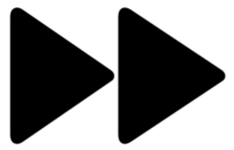


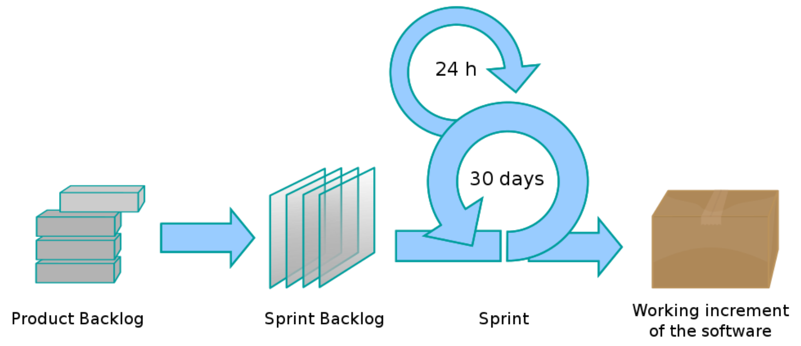
Scrum, Scrum, Scrum

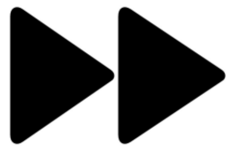


Scrum, Scrum, Scrum

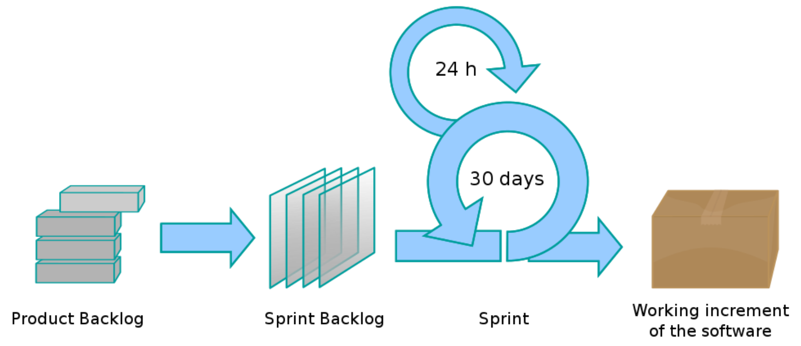


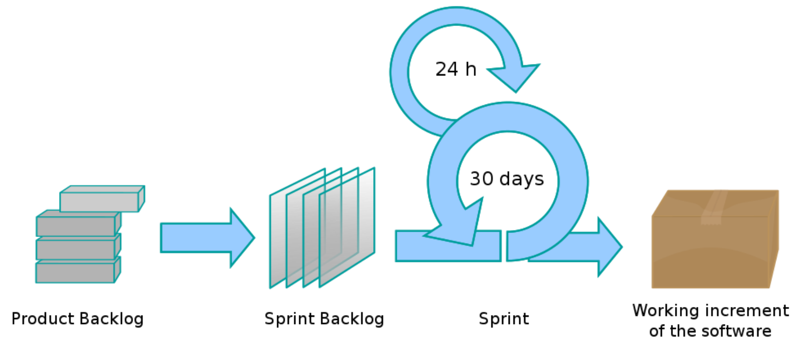


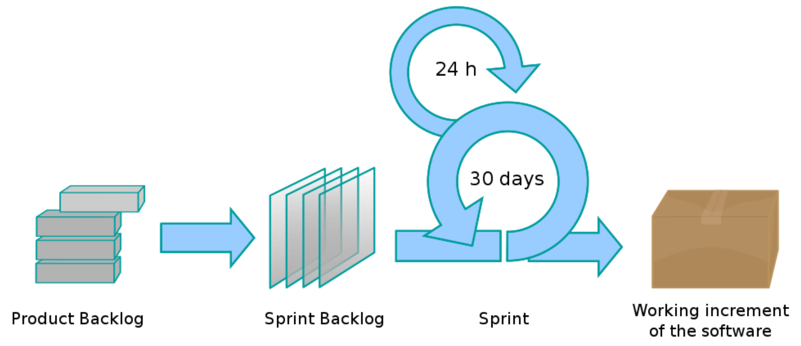




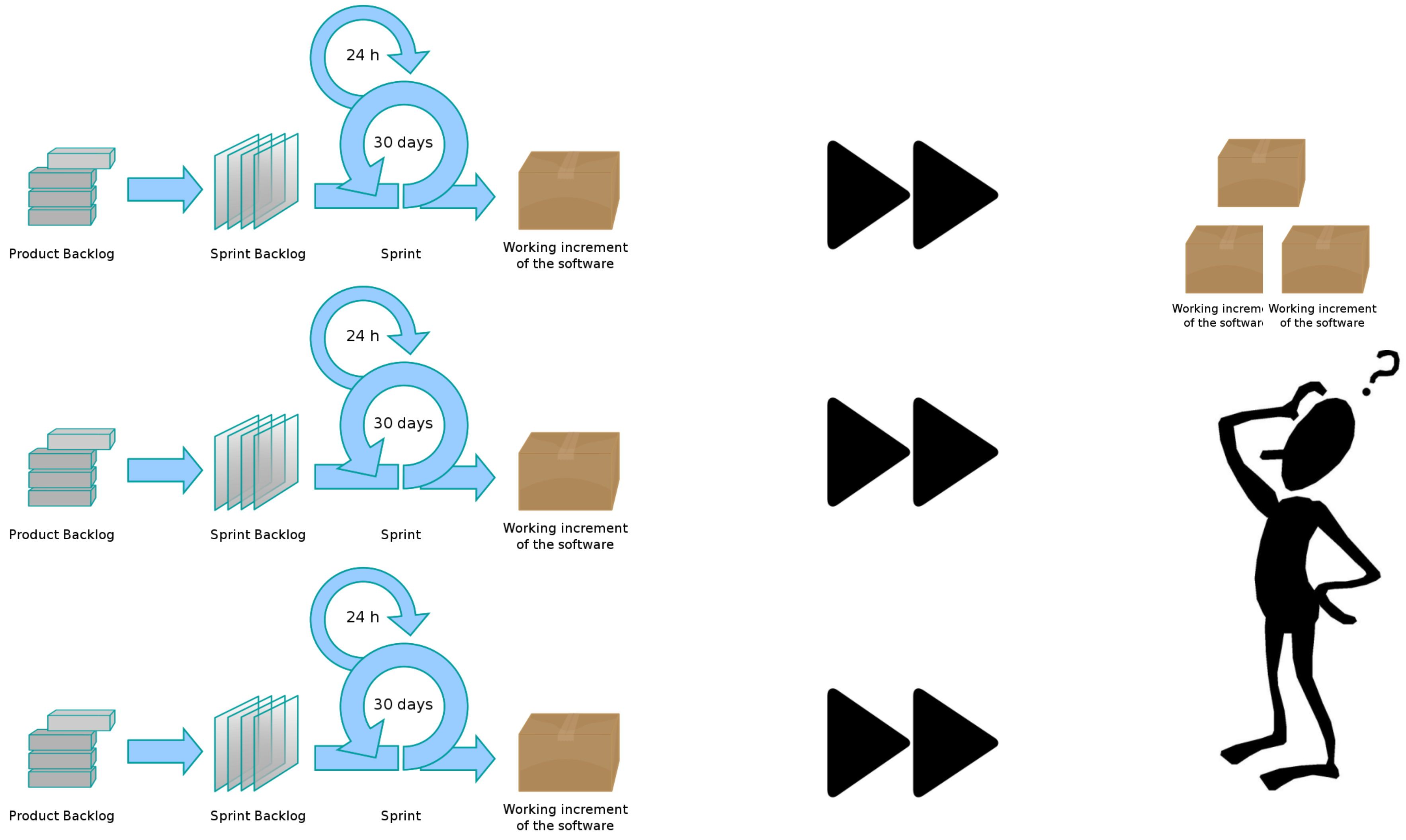
Scrum, Scrum, Scrum



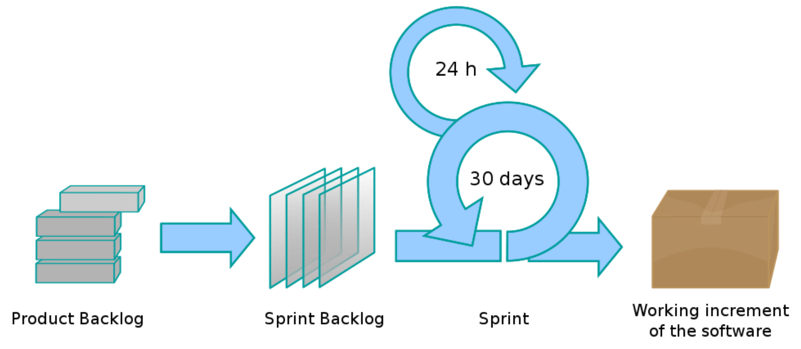


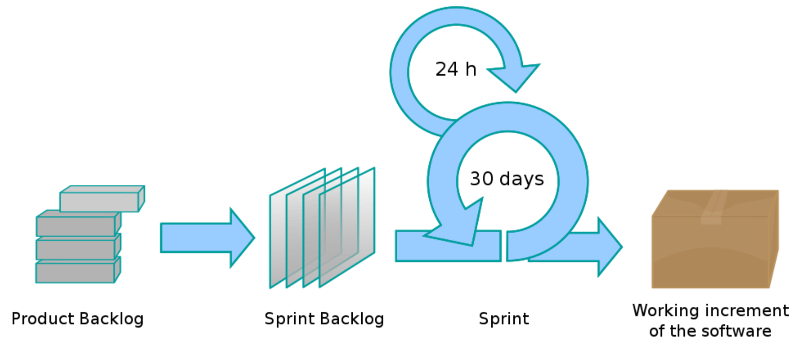


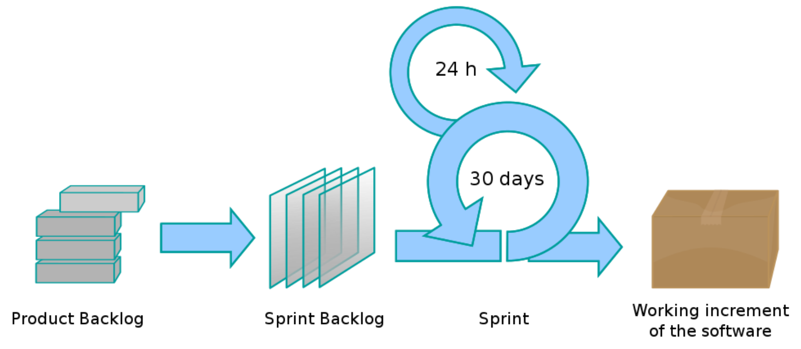
Scrum, Scrum, Scrum

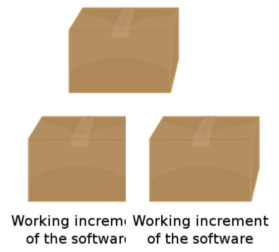

24 h
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment of the software
24 h
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment of the software
24 h
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment of the software
Working increm of the softwar
Working increment of the software

Scrum, Scrum, Scrum

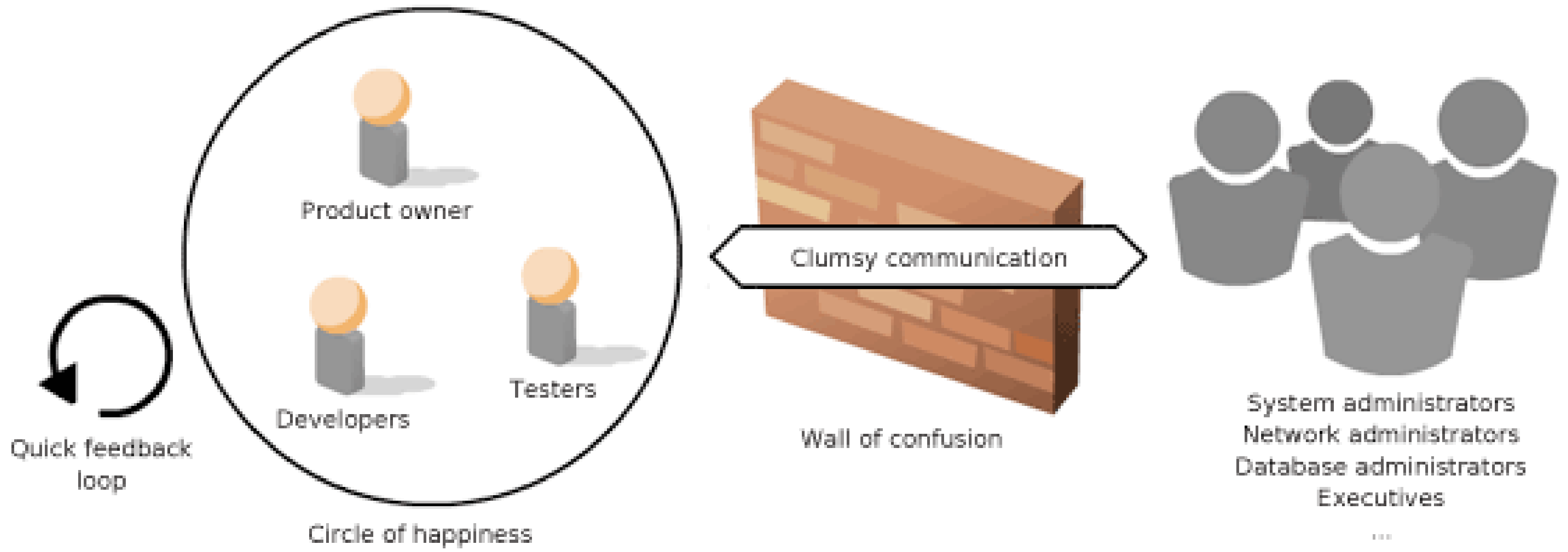




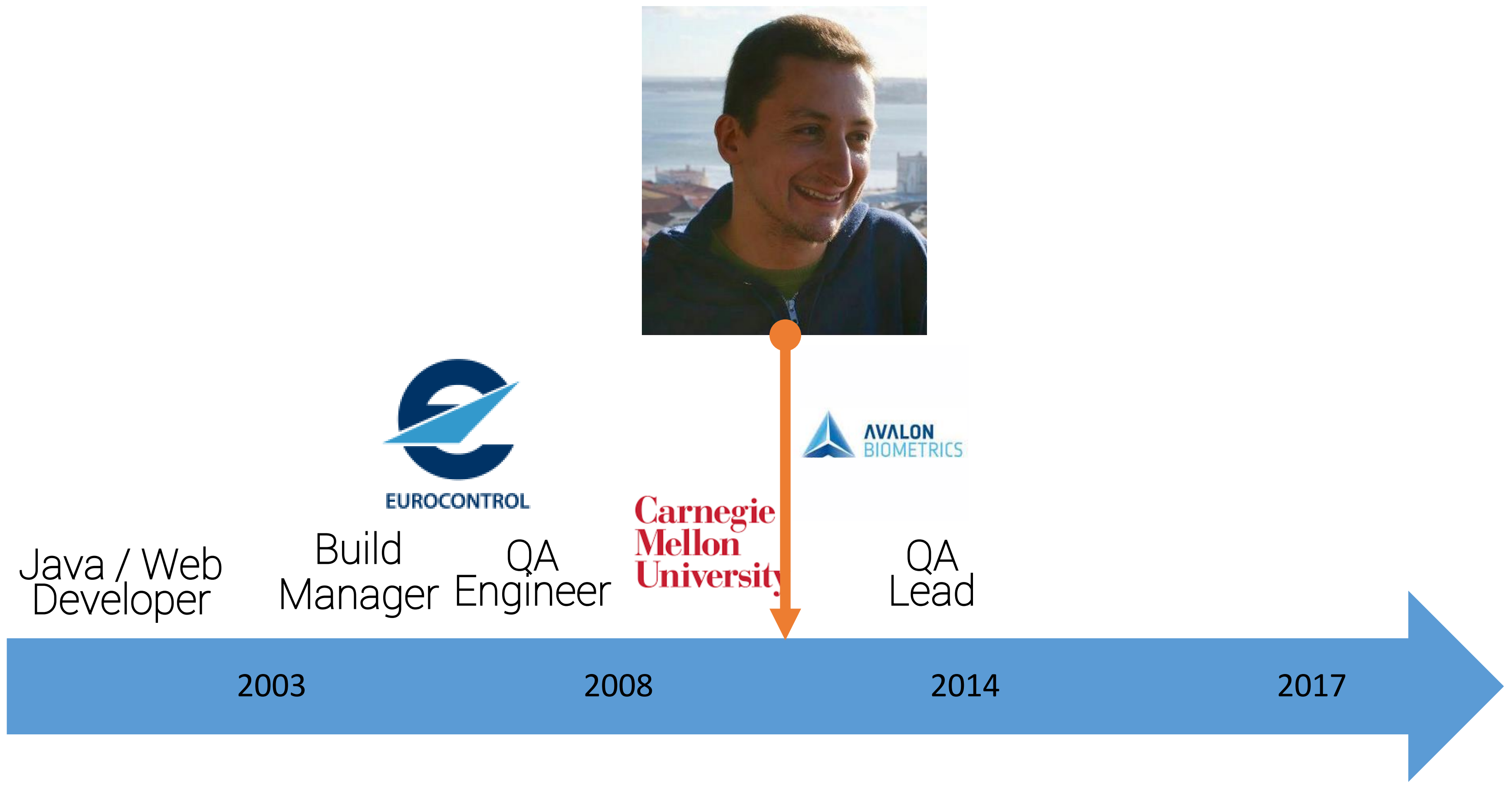




Wall of confusion



2009: DevOps


EUROCONTROL
Carnegie Mellon University
AVALON BIOMETRICS
Java / Web Developer
Build Manager
QA Engineer
QA Lead
2003
2008
2014
2017

“DevOps brought to the attention that two worlds, typically apart in a company, need to collaborate and that actually gives you a competitive edge”

–Patrick Debois

infoq.com/interviews/debois-devops

Vooza

WHAT IS ______________________________?

Culture

- Focus on People
- Embrace Change & experimentation

Automation

- “Continuous Delivery”
- “Infrastructure as Code”

Lean

- Focus on producing value for the end-user
- Small batch sizes

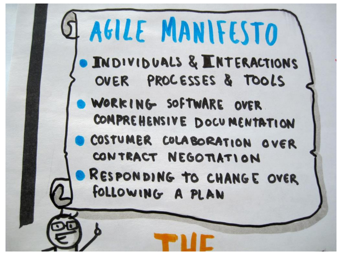
Measurement

- Measure everything
- Show the improvement

Sharing

- Open information sharing
- Collaboration & Communication

Agile... Scrum... DevOps

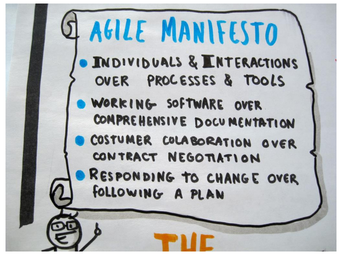


AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

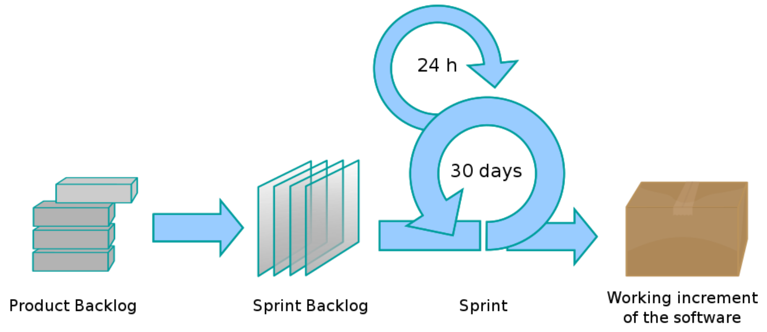
THE

Agile... Scrum... DevOps

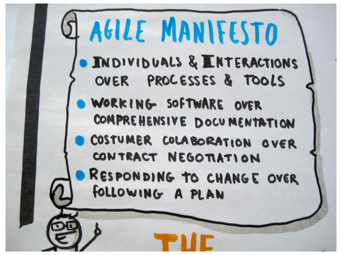

AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

THE



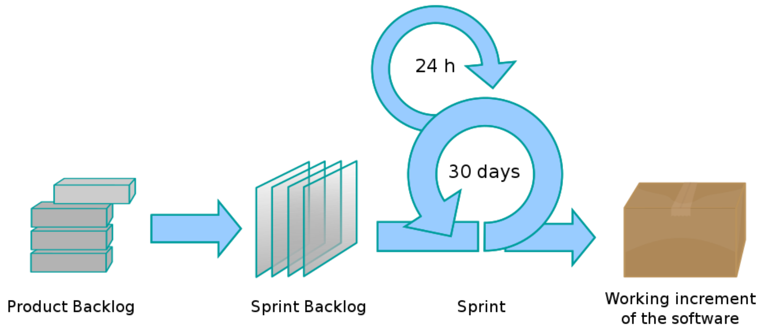
Agile… Scrum… DevOps



AGILE MANIFESTO

- INDIVIDUALS & INTERACTIONS OVER PROCESSES & TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- COSTUMER COLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

THE



“Agile System Administration was too long and too narrow… ”

–Patrick Debois

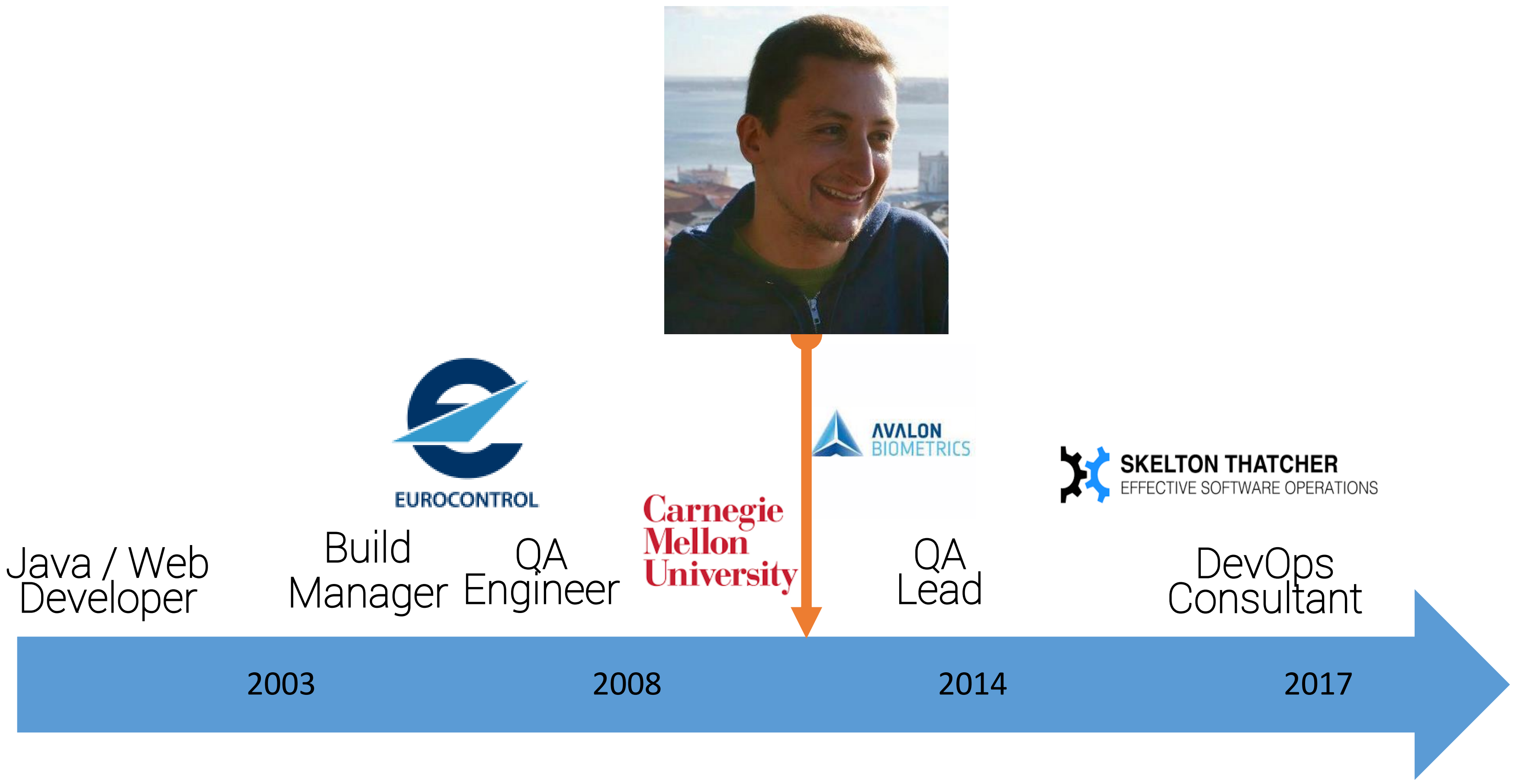
infoq.com/interviews/debois-devops

DevOps

From Wikipedia, the free encyclopedia

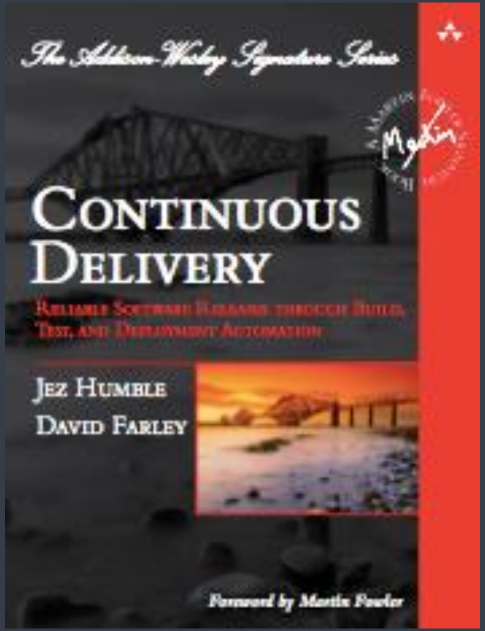
DevOps (a clipped compound of "development" and "operations") is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.[1][2] It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably.[3][4][5]

2010: Continuous Delivery

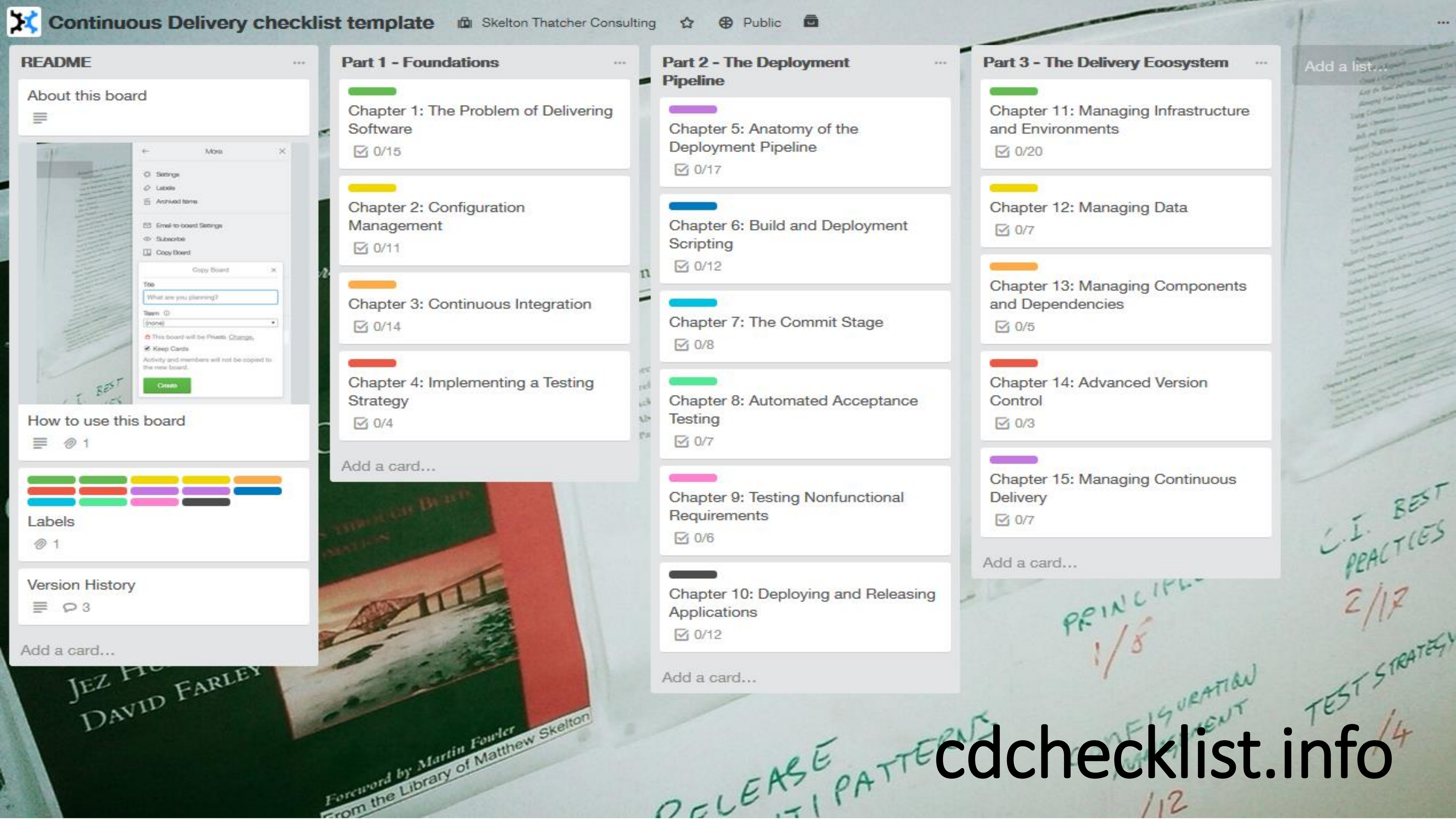


“ability to get changes of all types, into production, or into the hands of users, safely and quickly in a sustainable way”

–Jez Humble


The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

continuousdelivery.com



Continuous Delivery checklist template — Skelton Thatcher Consulting — Public

README

- About this board
- How to use this board — 1
- Labels — 1
- Version History — 3

Add a card…

Part 1 - Foundations

- Chapter 1: The Problem of Delivering Software — 0/15
- Chapter 2: Configuration Management — 0/11
- Chapter 3: Continuous Integration — 0/14
- Chapter 4: Implementing a Testing Strategy — 0/4

Add a card…

Part 2 - The Deployment Pipeline

- Chapter 5: Anatomy of the Deployment Pipeline — 0/17
- Chapter 6: Build and Deployment Scripting — 0/12
- Chapter 7: The Commit Stage — 0/8
- Chapter 8: Automated Acceptance Testing — 0/7
- Chapter 9: Testing Nonfunctional Requirements — 0/6
- Chapter 10: Deploying and Releasing Applications — 0/12

Add a card…

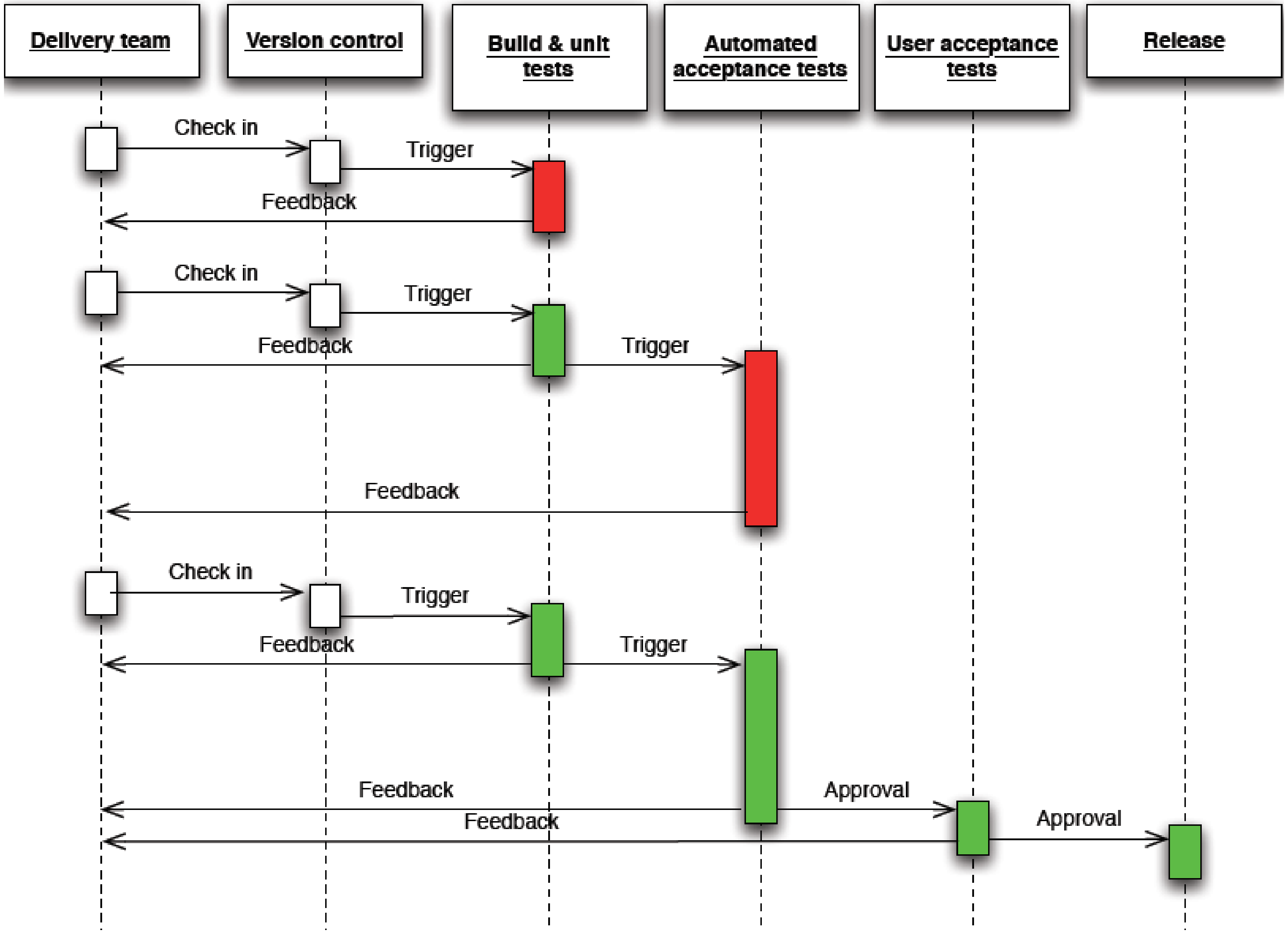
Part 3 - The Delivery Ecosystem

- Chapter 11: Managing Infrastructure and Environments — 0/20
- Chapter 12: Managing Data — 0/7
- Chapter 13: Managing Components and Dependencies — 0/5
- Chapter 14: Advanced Version Control — 0/3
- Chapter 15: Managing Continuous Delivery — 0/7

Add a card…

Add a list…

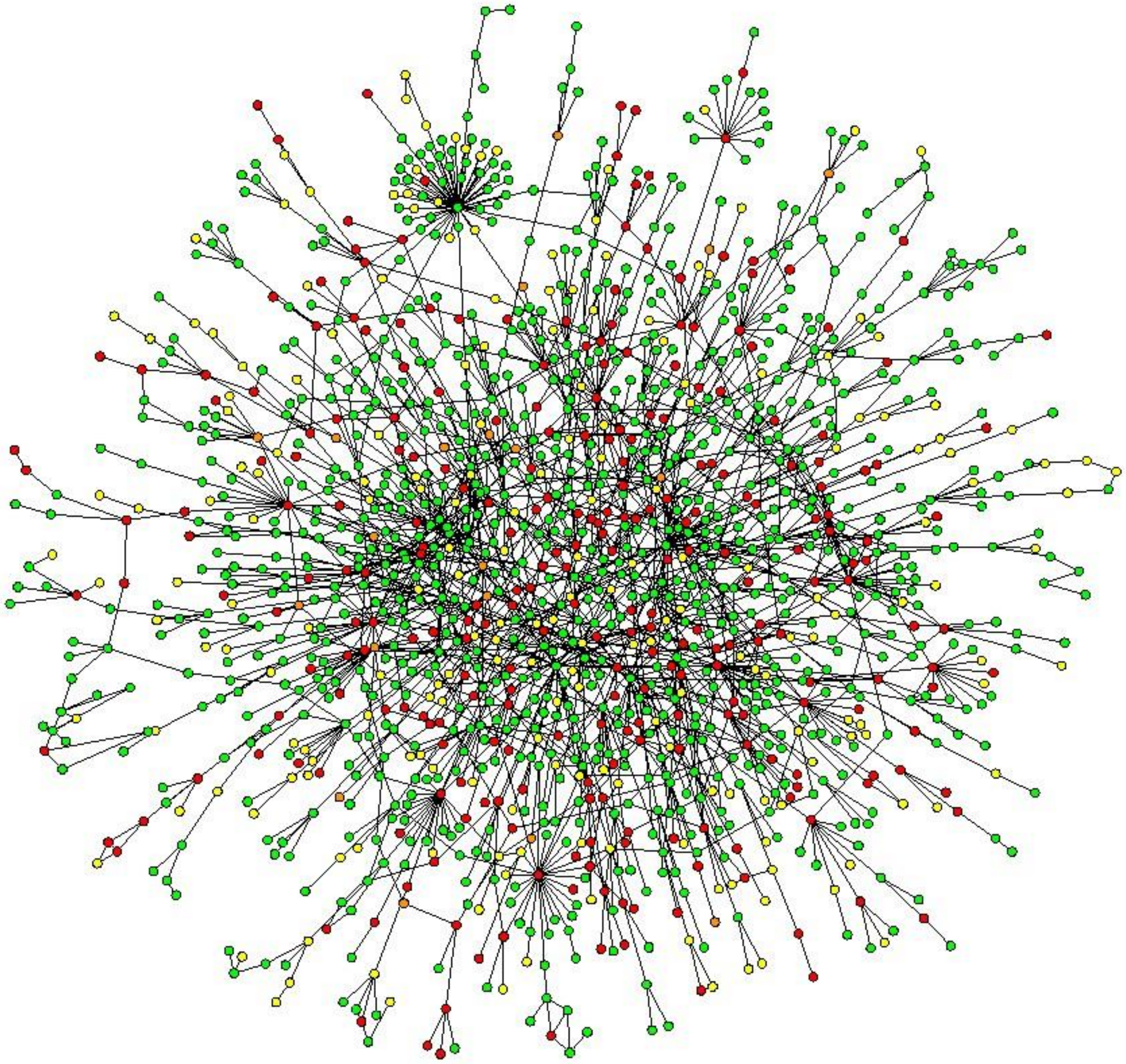
cdchecklist.info

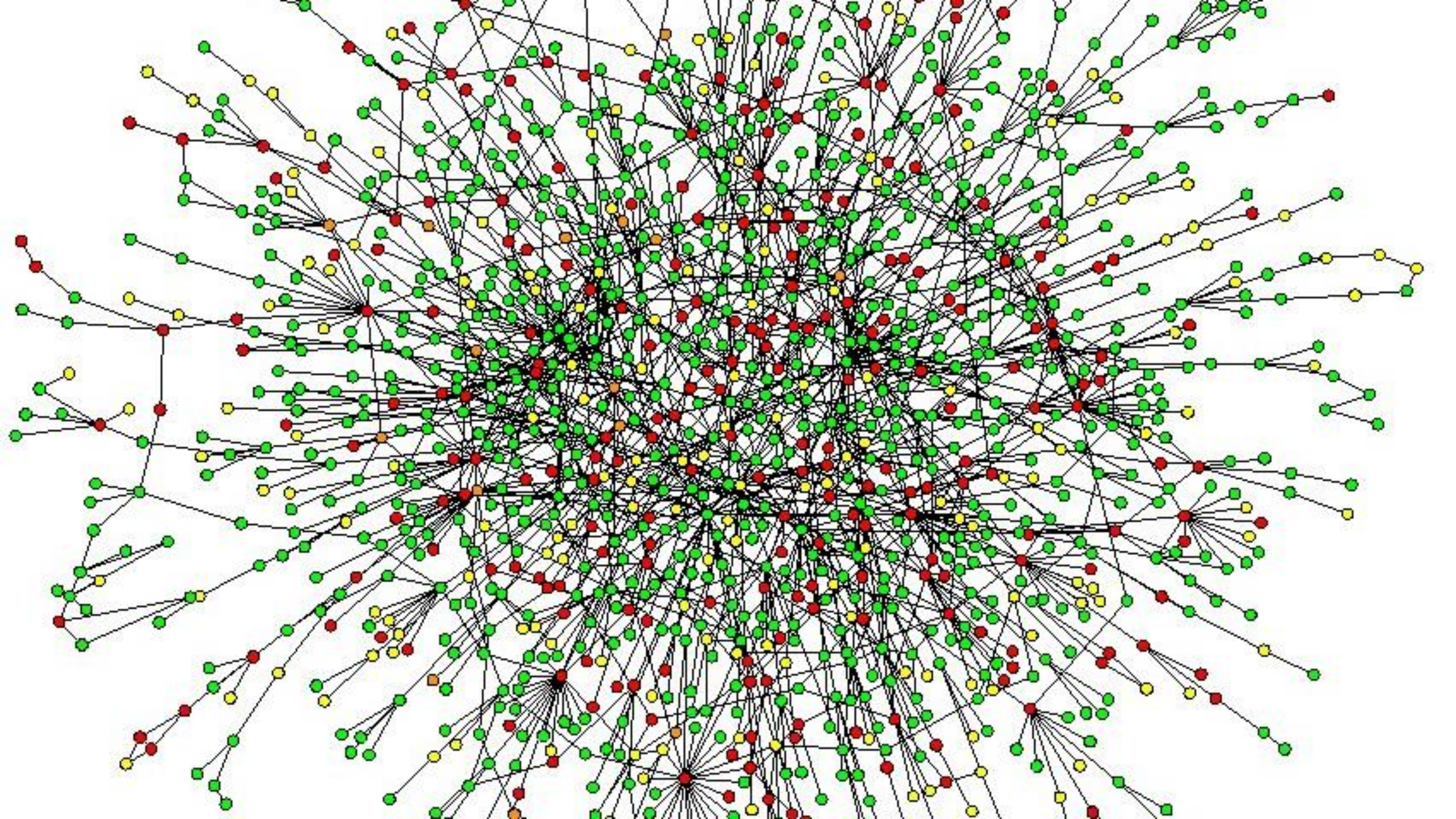


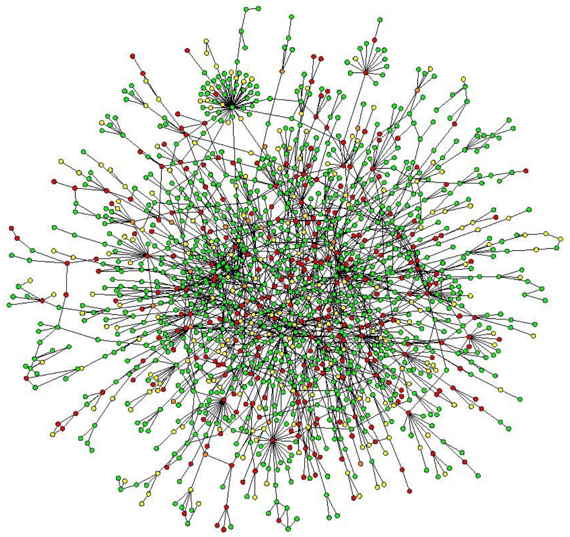
Credits: Jez Humble, Martin Fowler, Tom Sulston, Sam Newman

Agenda

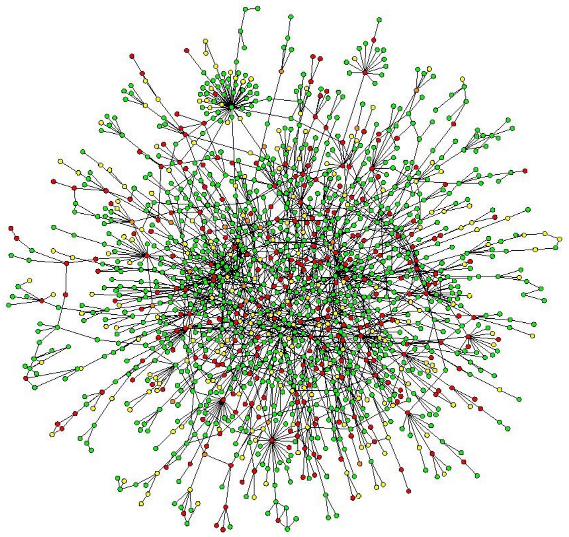
1. The Need for Speed (aka DevOps)
2. Failure = Quality in Complex Systems
3. Survival of High-Performing Cultures

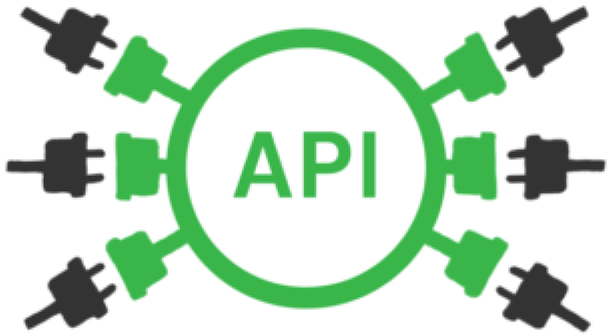


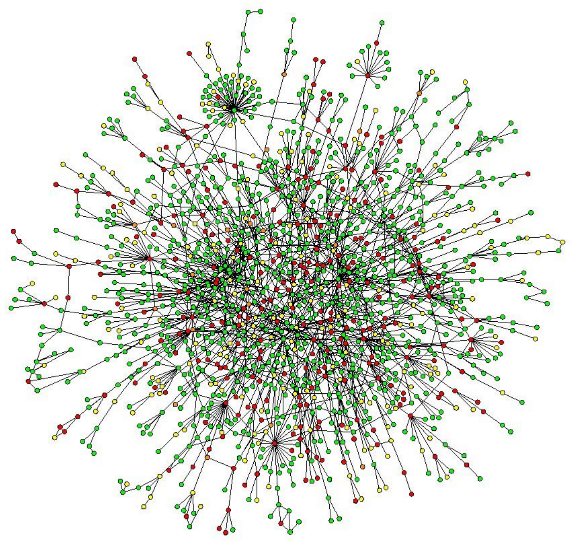


OPEN
SOFTWARE
PUBLIC
COPYRIGHT
COST
ACCESS
TECHNOLOGY
MOVEMENT
OPEN-SOURCE
SYSTEM
USE
COMPUTER
DEVELOPMENT
COMMUNITY
DIGITAL
MEDIA
INTERNET
TERMS
SYSTEMS
MANY
MADE
EXAMPLE
CONTENT
INFORMATION
OPERATING
FREE
SOURCE
AVAILABLE
RESEARCH
CODE
CULTURE
CREATIVE
USERS
CULTURAL
SHARING
PRODUCTION


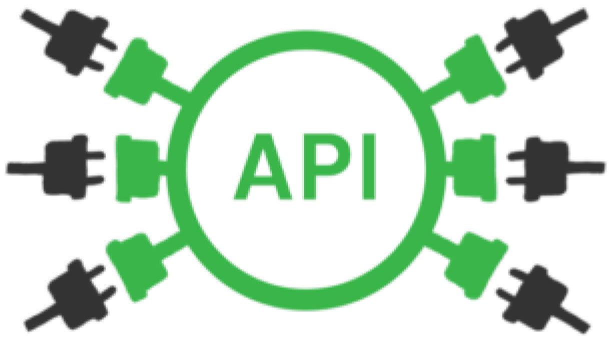
OPEN
PUBLIC
COPYRIGHT
ACCESS
SOFTWARE
TECHNOLOGY
MOVEMENT
OPEN-SOURCE
SYSTEM
COMMUNITY
DIGITAL
DEVELOPMENT
MEDIA
INTERNET
CONTENT
INFORMATION
FREE
SOURCE
AVAILABLE
RESEARCH
CODE
CULTURE
USERS
CULTURAL

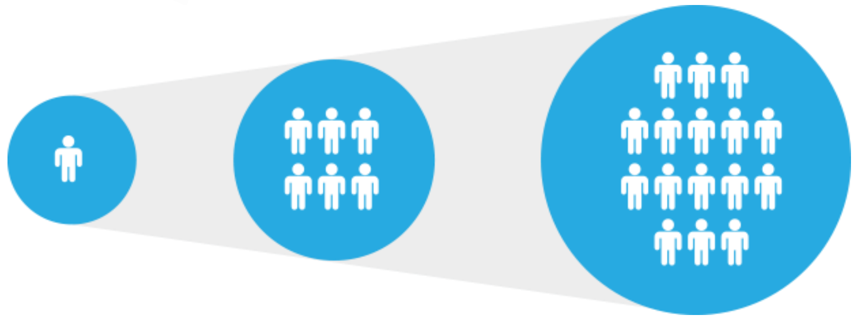


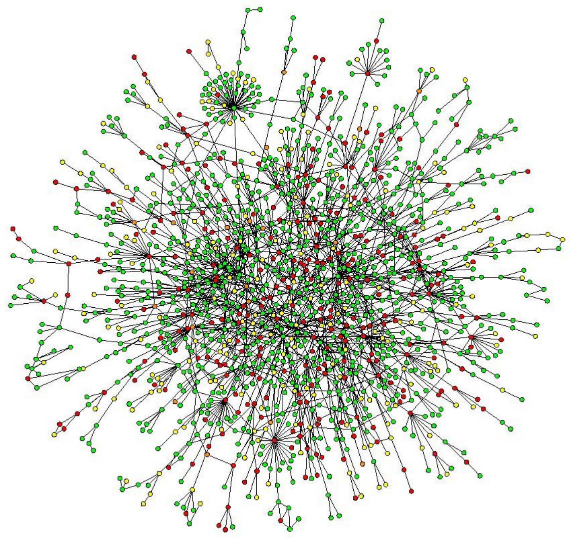

API



OPEN
SOFTWARE
PUBLIC
COPYRIGHT
ACCESS
TECHNOLOGY
MOVEMENT
OPEN-SOURCE
COMMUNITY
DIGITAL
DEVELOPMENT
INTERNET
MEDIA
INFORMATION
CONTENT
FREE
SOURCE
AVAILABLE
RESEARCH
CODE
CULTURE
USERS
CULTURAL


API





Systems of systems

Complex run time dependencies

Vulnerable build time dependencies

Failure is endemic

Everybody right now.
#AWS #awscloud #awsoutage #awsdown #S3
#AWSs3 #Amazon

RETWEETS 443 LIKES 592

11:38 AM - 28 Feb 2017

Everybody right now.
#AWS #awscloud #awsoutage #awsdown #S3
#AWSs3 #Amazon

RETWEETS 443 LIKES 592

11:38 AM - 28 Feb 2017

DevOps

GitHub: We're sorry (again) about (another) outage

Sky blue, oceans wet, code sharer unstable

By Shaun Nichols in San Francisco 29 Jan 2016 at 19:27 17 SHARE ▼

Everybody right now.
#AWS #awscloud #awsoutage #awsdown #S3
#AWSs3 #Amazon

RETWEETS 443 LIKES 592

11:38 AM - 28 Feb 2017

DevOps

GitHub: We're sorry (again) about (another) outage

Sky blue, oceans wet, code sharer unstable

By Shaun Nichols in San Francisco 29 Jan 2016 at 19:27 17 SHARE ▼

News › Business › Business News

British Airways system outage 'caused by IT worker accidentally switching off power supply'

“The zero-error fallacy”

Researchers at MIT have shown that:

a) the more incidents an airline has, the lower the passenger mortality risk

https://techbeacon.com/zero-error-fallacy-what-really-counts-devops-teams

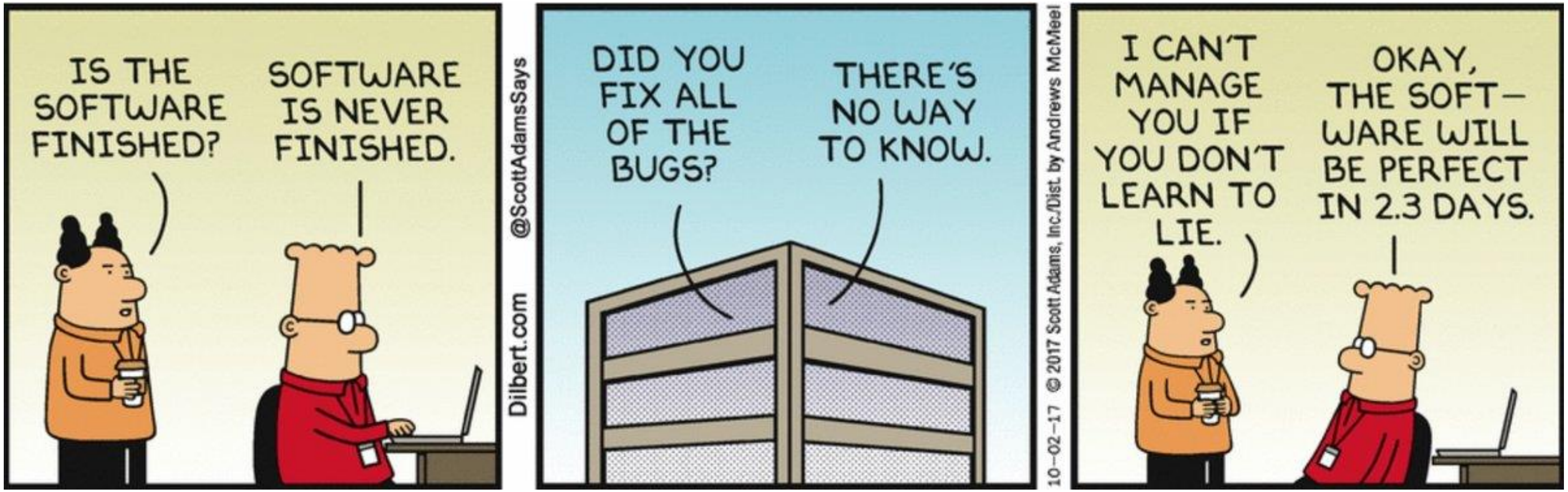
“The zero-error fallacy”

Researchers at MIT have shown that:

a) the more incidents an airline has, the lower the passenger mortality risk

b) construction sites with relatively more incidents in a given year have fewer worker deaths than those with zero incidents.

https://techbeacon.com/zero-error-fallacy-what-really-counts-devops-teams


IS THE SOFTWARE FINISHED?
SOFTWARE IS NEVER FINISHED.
Dilbert.com @ScottAdamsSays
DID YOU FIX ALL OF THE BUGS?
THERE'S NO WAY TO KNOW.
10-02-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
I CAN'T MANAGE YOU IF YOU DON'T LEARN TO LIE.
OKAY, THE SOFT-WARE WILL BE PERFECT IN 2.3 DAYS.

Learning from Failure

Greatest illusion is that the difference between excellent and crappy operations is the number of errors or failures

Learning from Failure

Greatest illusion is that the difference between excellent and crappy operations is the number of errors or failures

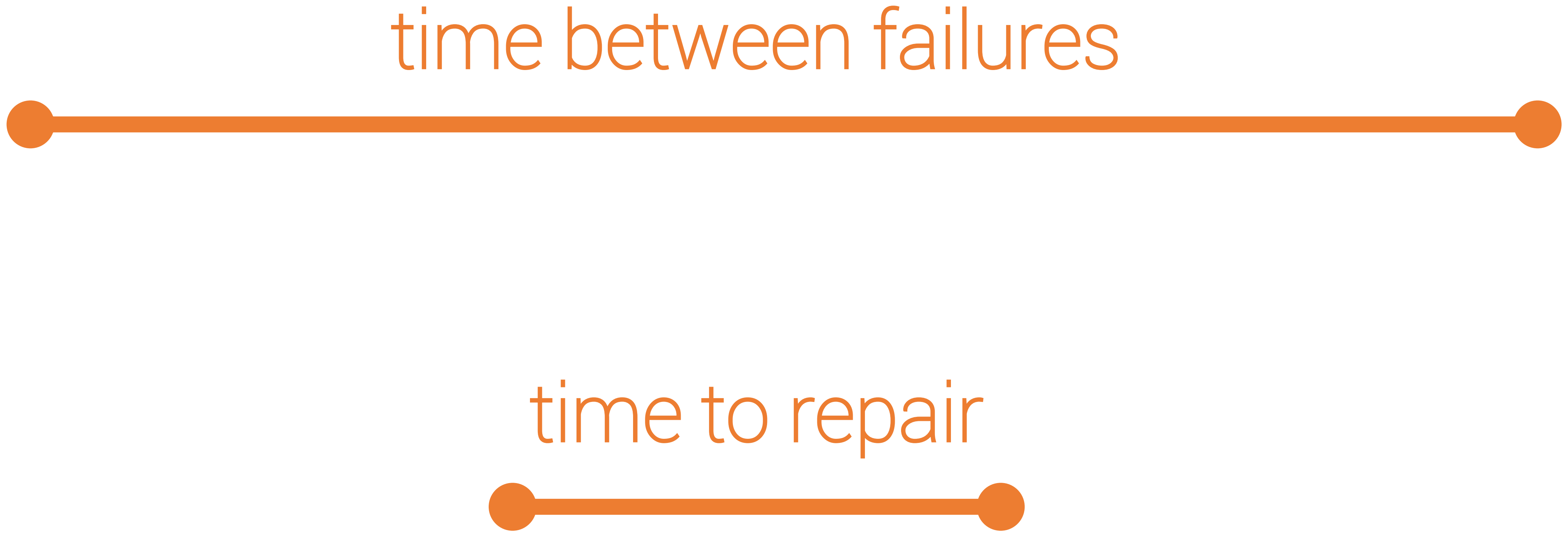
What makes a difference is the presence of positive capacities—in people, in teams, in the organization.

Learning from Failure

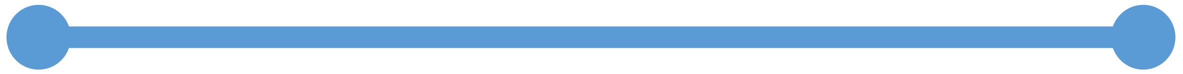
Greatest illusion is that the difference between excellent and crappy operations is the number of errors or failures

What makes a difference is the presence of positive capacities—in people, in teams, in the organization.

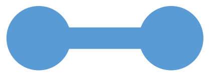
A safety culture is one in which the boss actually invites bad news, and may even reward it.


time between failures
time to repair

time between failures



time to repair



Development vs Maintenance

~~Development vs Maintenance~~

TEAM builds, deploys, runs, monitors and fixes
+ Ops provides platform

DONE = deployed

~~DONE = deployed~~

DONE = monitored in production

DONE = deployed

DONE = ~~monitored~~ NOT in production

Incident reviews

It's Not Your Fault

(Blameless)

post-mortems

@jasonhand

http://www.slideshare.net/jhand2/its-not-your-fault-blameless-post-mortems

Chaos engineering

https://medium.com/netflix-techblog/chaos-engineering-upgraded-878d341f15fa

Wrong incentives

Rewarding zero defects / fixing defects

Wrong incentives

Rewarding zero defects / fixing defects

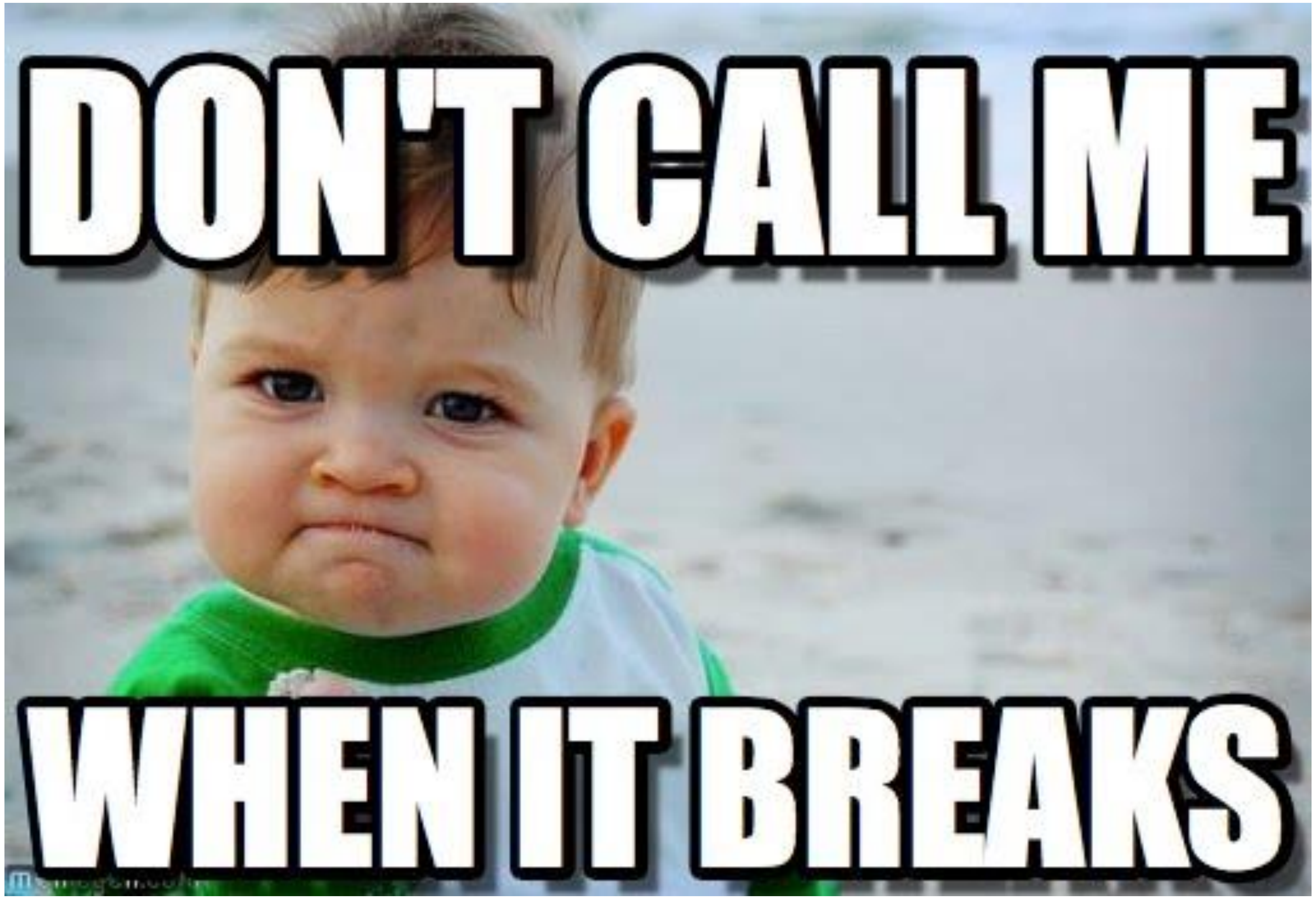
Focus on simple/single metric

Wrong incentives

Rewarding zero defects / fixing defects

Focus on simple/single metric

Different IT teams with different goals


DON'T CALL ME
WHEN IT BREAKS

Right incentives

Reward along business objectives

Right incentives

Reward along business objectives

Combination of metrics (e.g. lead time + time to repair)

Right incentives

Reward along business objectives

Combination of metrics (e.g. lead time + time to repair)

All IT teams share same objectives

High Performers Are More Agile

30x

more frequent deployments

200x

faster lead times than their peers

Source: Puppet Labs 2015 State Of DevOps: https://puppetlabs.com/2015-devops-report

High Performers Are More Reliable

60x

the change success rate

168x

faster mean time to recover (MTTR)

Source: Puppet Labs 2015 State Of DevOps: https://puppetlabs.com/2015-devops-report

Agenda

1. The Need for Speed (aka DevOps)
2. Failure = Quality in Complex Systems
3. Survival of High-Performing Cultures

Culture Types

Pathological (power-oriented)	Bureaucratic (rule-oriented)	Generative (performance-oriented)
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to enquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

source: http://continuousdelivery.com/implementing/culture

Pathological (power-oriented)
Low cooperation
Messengers shot
Responsibilities shirked
Bridging discouraged
Failure leads to scapegoating
Novelty crushed

Blame Culture

Pathological (power-oriented)
Low cooperation
Messengers shot
Responsibilities shirked
Bridging discouraged
Failure leads to scapegoating
Novelty crushed

Blame Culture

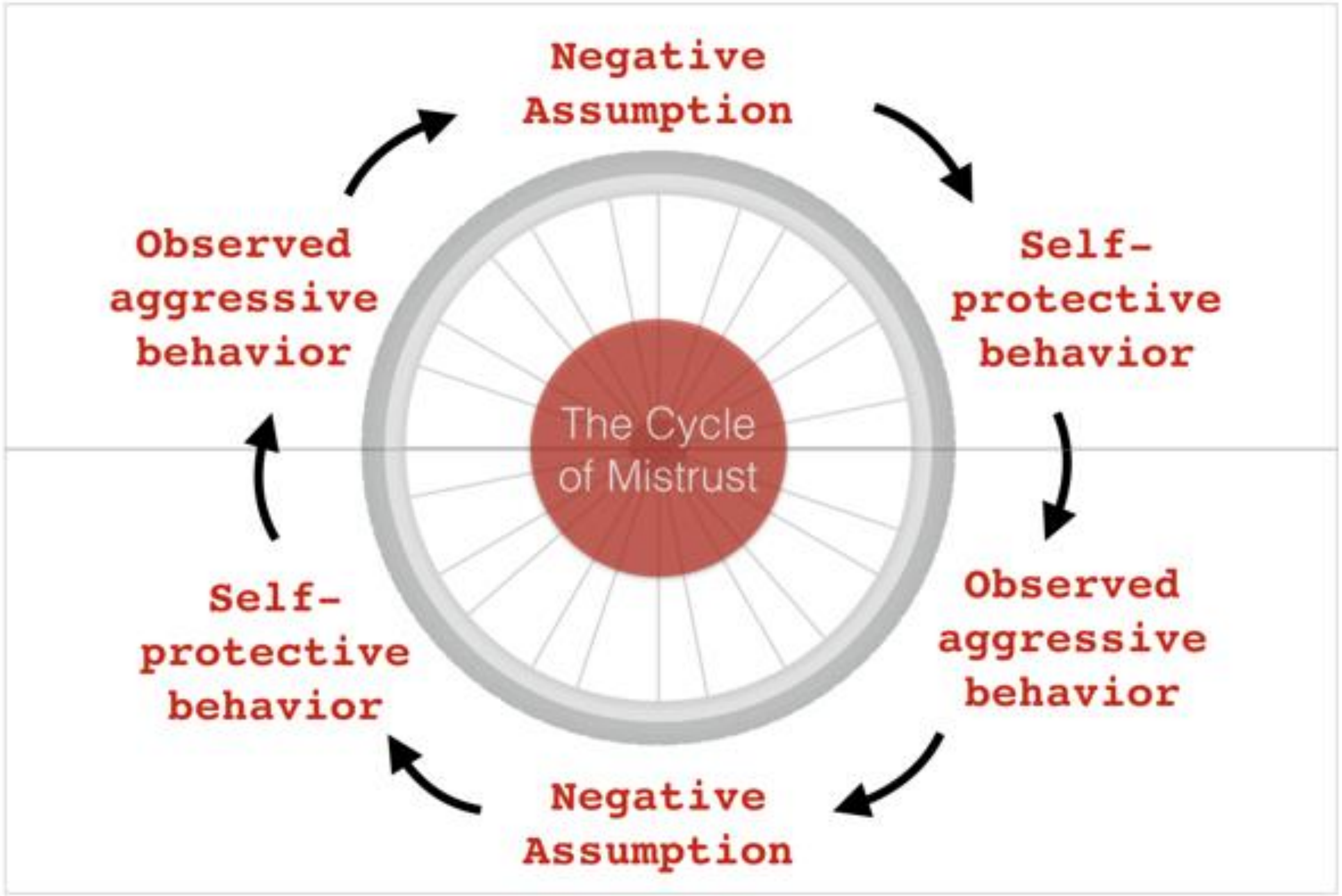
Resistance to Change

Pathological (power-oriented)
Low cooperation
Messengers shot
Responsibilities shirked
Bridging discouraged
Failure leads to scapegoating
Novelty crushed

Blame Culture

Resistance to Change

Lack of Collaboration



Designed by Joshua Kerievsky. Adapted from "Driving Fear Out of the Workplace"

Blameless

Generative (performance-oriented)
High cooperation
Messengers trained
Risks are shared
Bridging encouraged
Failure leads to enquiry
Novelty implemented

Blameless

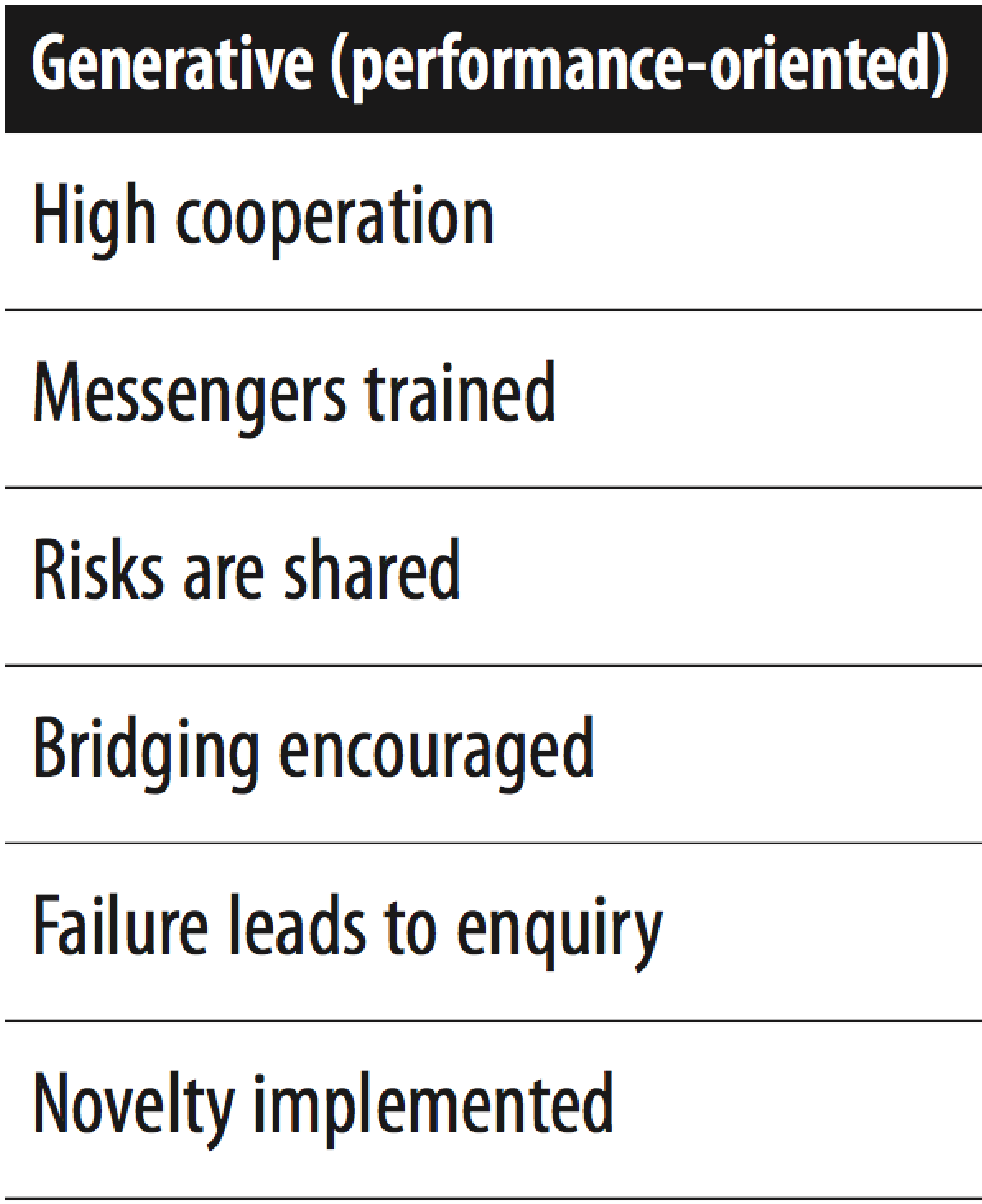
Continuous Learning

Generative (performance-oriented)
High cooperation
Messengers trained
Risks are shared
Bridging encouraged
Failure leads to enquiry
Novelty implemented

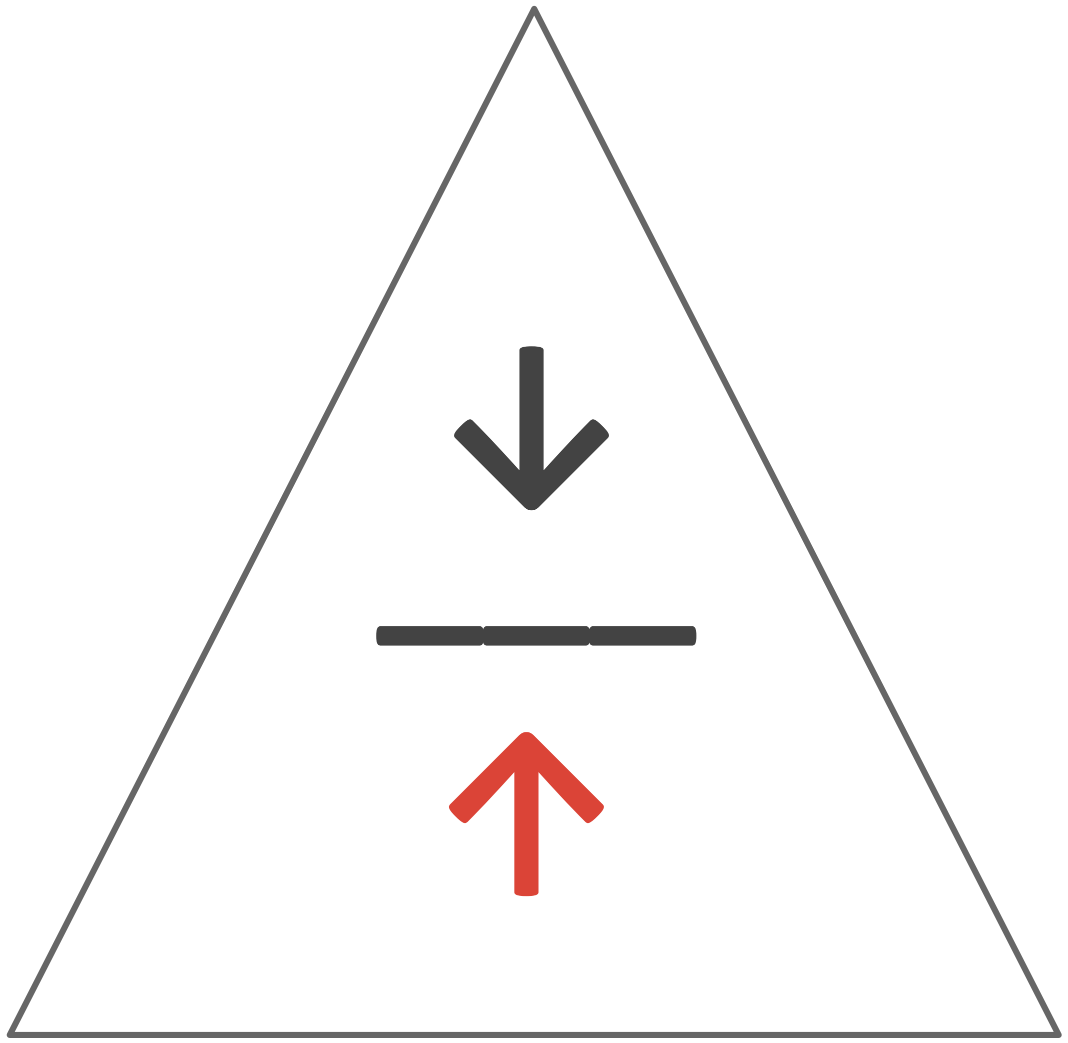
Blameless

Continuous Learning

High Collaboration

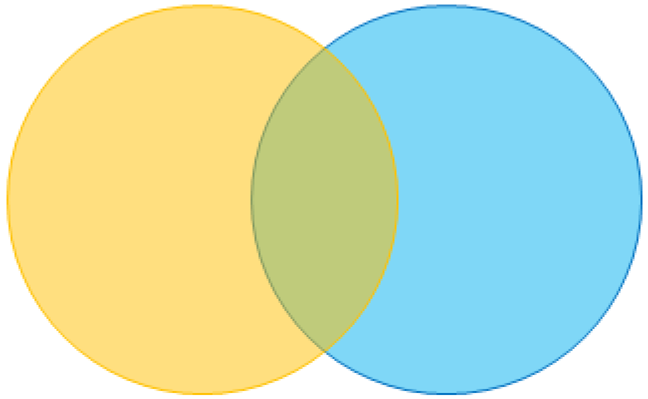


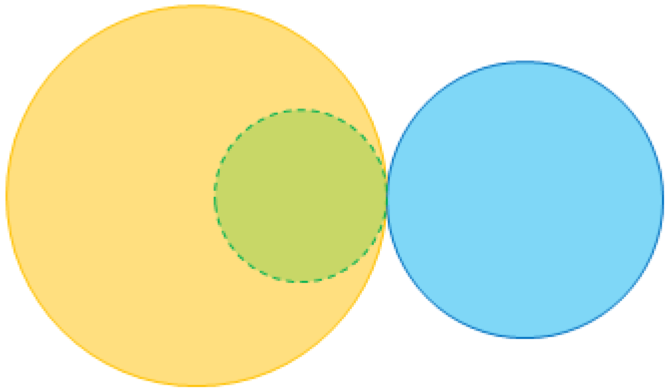
Generative (performance-oriented)
High cooperation
Messengers trained
Risks are shared
Bridging encouraged
Failure leads to enquiry
Novelty implemented



http://itrevolution.com/speeding-scaling-devops-enterprise

Collaboration vs X-as-a-Service

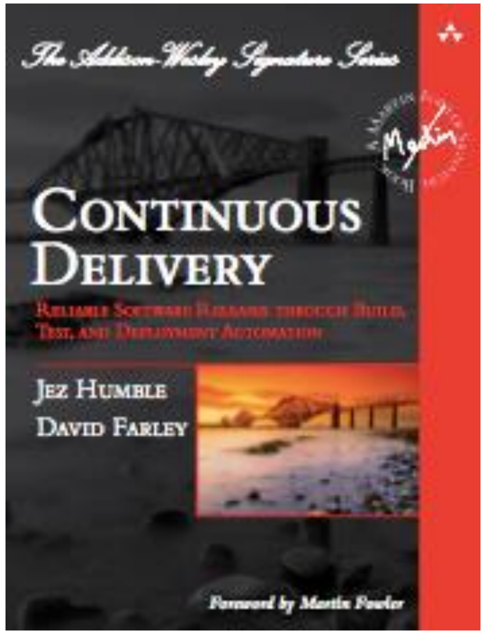




devopstopologies.com

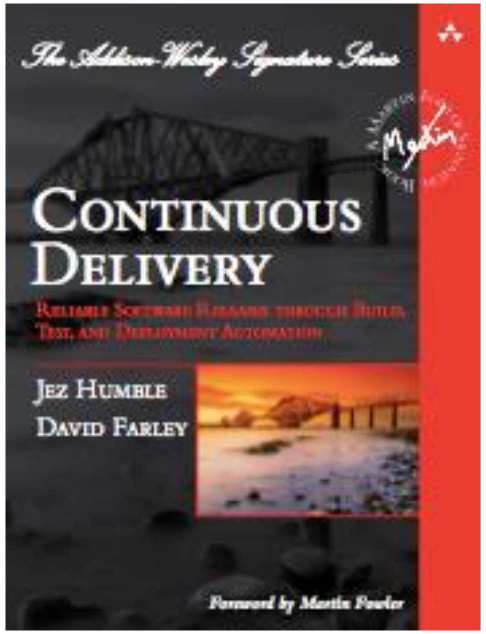
	Collaboration		X-as-a-Service
+	Rapid discovery No hand-offs	+	Ownership clarity Less context needed
−	Comms overheads?	−	Slower innovation?

Conclusion


The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

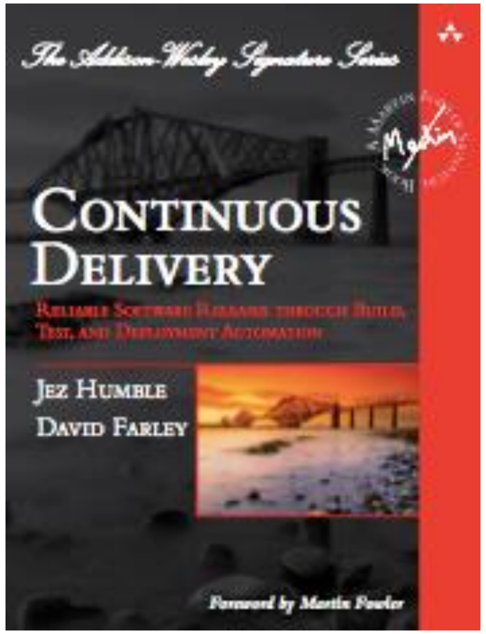
Generative (performance-oriented)
High cooperation
Messengers trained
Risks are shared
Bridging encouraged
Failure leads to enquiry
Novelty implemented

Conclusion


The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

People

Conclusion



The Addison-Wesley Signature Series

Continuous Delivery

Reliable Software Releases through Build, Test, and Deployment Automation

Jez Humble

David Farley

Foreword by Martin Fowler

Process

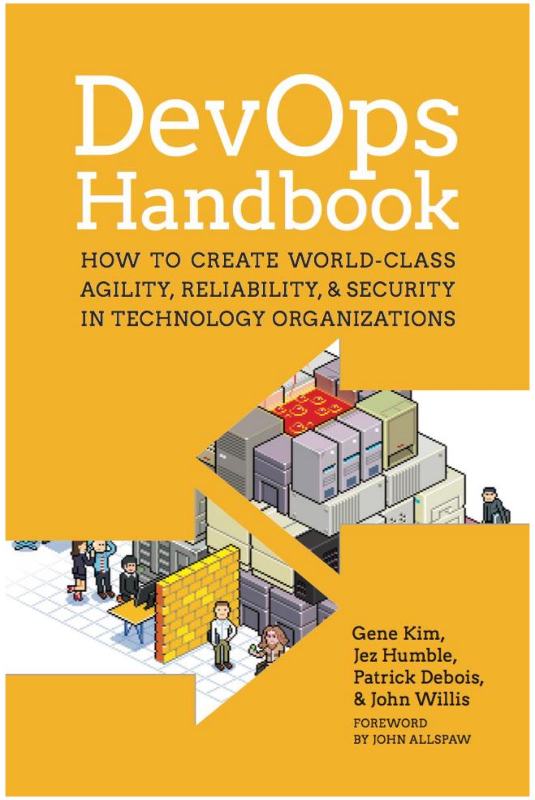
People

Conclusion

Tools

Process

People

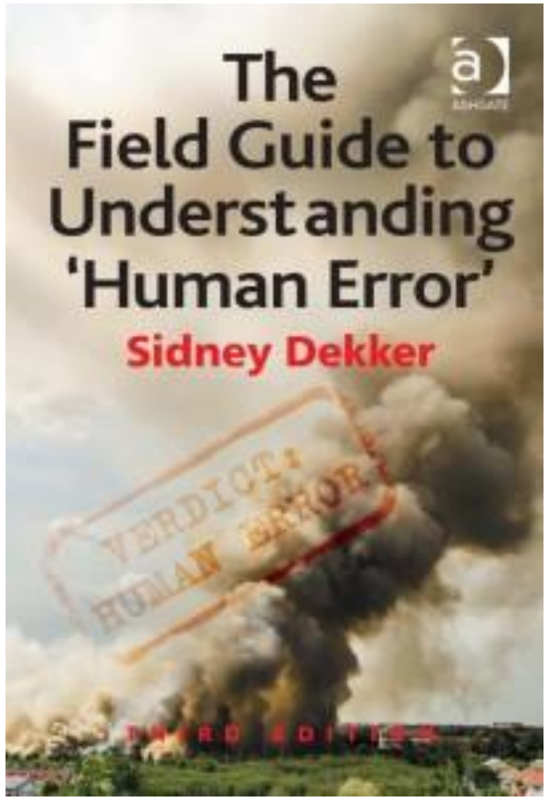


DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS

Gene Kim, Jez Humble, Patrick Debois, & John Willis

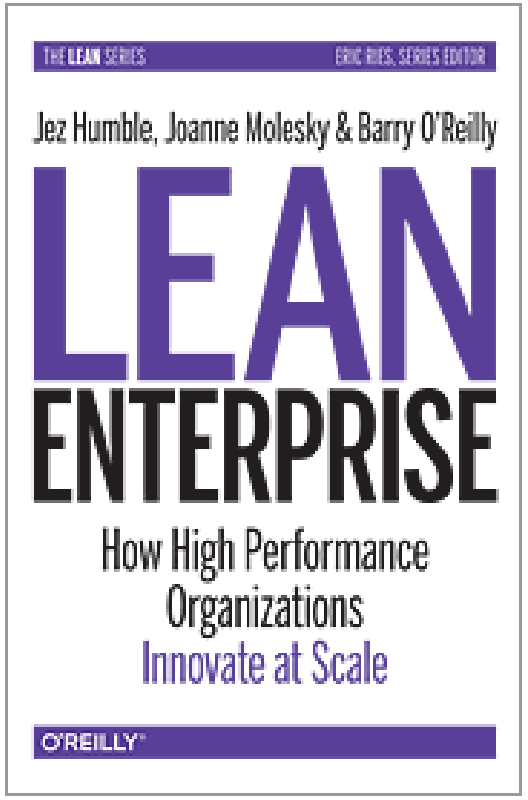
FOREWORD BY JOHN ALLSPAW



The Field Guide to Understanding 'Human Error'

Sidney Dekker

THIRD EDITION



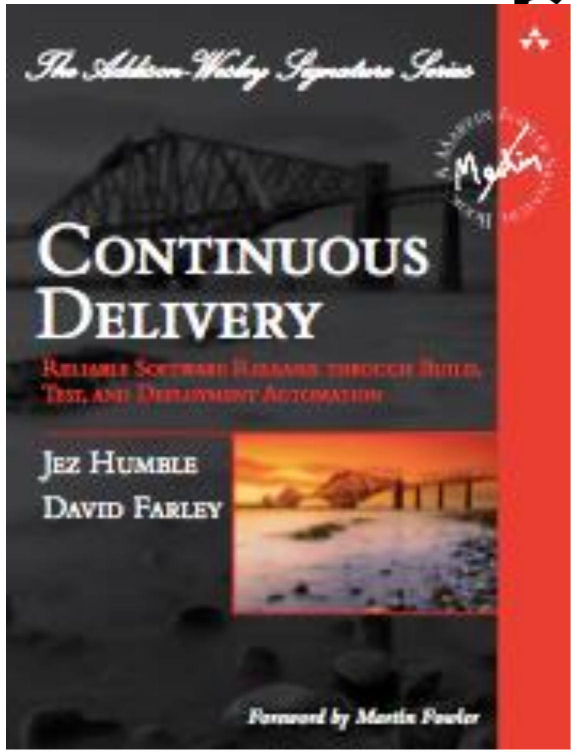
THE LEAN SERIES ERIC RIES, SERIES EDITOR

Jez Humble, Joanne Molesky & Barry O'Reilly

LEAN ENTERPRISE

How High Performance Organizations Innovate at Scale

O'REILLY



The Addison-Wesley Signature Series

CONTINUOUS DELIVERY

RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION

JEZ HUMBLE DAVID FARLEY

Foreword by Martin Fowler

References

Thank you!

Manuel Pais

MS Software Eng Carnegie Mellon University

@manupaisable

manuelpais.net

manuel.pais@gmail.com

DevOps and Delivery Consultant

Focused on teams and flow