



 DevOps Institute

# DevOps Foundation®

01Mar2021

# Tell Us a Little About Yourself

Please let us know who you are:

- Name, organization and role
- DevOps/Agile/Lean/ITSM experience
- Why you are attending this course
- What you expect to learn



What is your definition or perception of DevOps?

# DevOps Foundation Course Goals

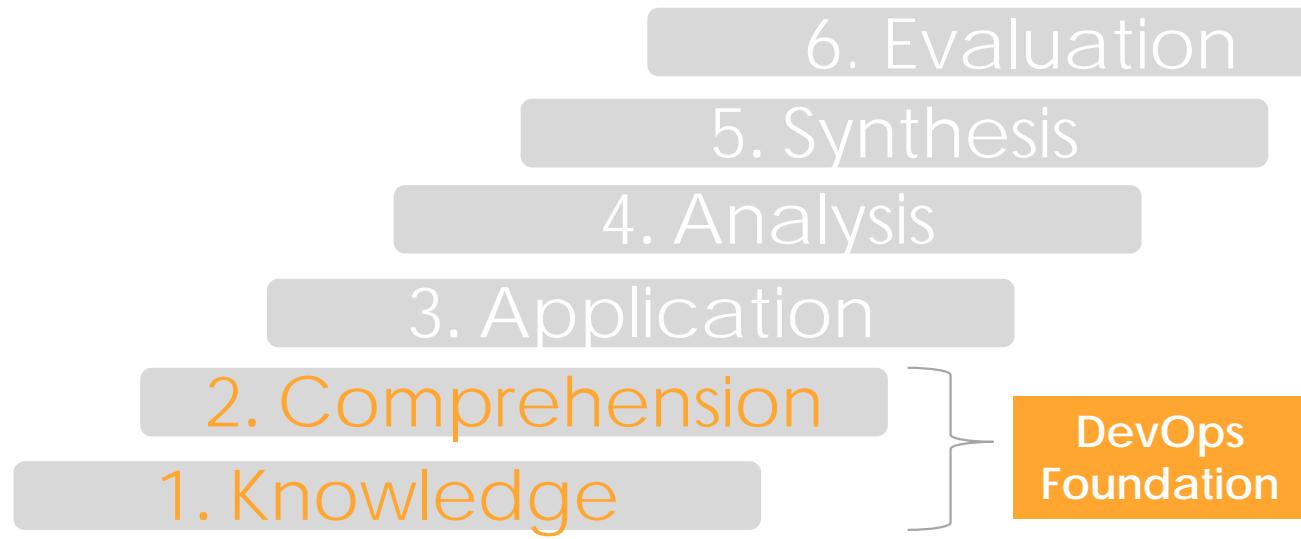
- Learn about DevOps
- Understand its core vocabulary, principles, practices and automation
- Hear and share real life scenarios
- Have fun!



Pass the DevOps Foundation Exam

- 40 multiple choice questions
- 60 minutes
- 65% is passing
- Accredited by DevOps Institute
- Get your digital badge

# About Bloom's Taxonomy



Bloom's Taxonomy is used to categorize learning objectives and, from there, assess learning achievements.

# About DevOps Institute



DevOps Institute is dedicated to advancing the human elements of DevOps success. As a global member association, DevOps Institute is the go-to hub connecting IT practitioners, industry thought leaders, talent acquisition, business executives and education partners to help pave the way to support digital transformation and the New IT.

DevOps Institute helps advance careers and professional development within the DevOps community through recognized certifications, research and thought leadership, events and the fastest-growing DevOps member community.

# DevOps Foundation Course Content

Day 1	Day 2
Hello! Course & Class Welcome	
Module 1 Exploring DevOps	Module 5 <b>Culture</b> , Behaviors and Operating Models
Module 2 Core DevOps Principles	Module 6 <b>Automation</b> & Architecting DevOps Toolchains
Module 3 Key DevOps Practices	Module 7 <b>Measurement</b> , Metrics & Reporting
Module 4 Business and Technology Frameworks	Module 8 <b>Sharing</b> , Shadowing & Evolving
Sample Examination Review	Examination

# Module 1

# EXPLORING DEVOPS

# Module 1: Exploring DevOps

- Defining DevOps
- Why Does DevOps Matter
- The Business Perspective
- The IT Perspective

Component	Module 1 Content
Video	A Short History of DevOps with Damon Edwards
Case Story	ING Bank, Netherlands
Discussion	DevOps Myths versus Realities
Exercise	Your Organizational Why

# Defining DevOps

Module 1: Exploring DevOps

9

©DevOps Institute unless otherwise stated

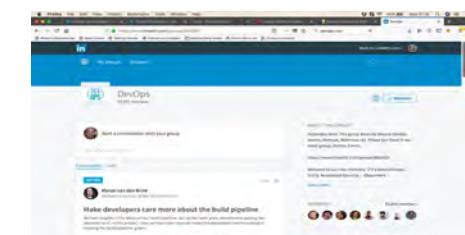
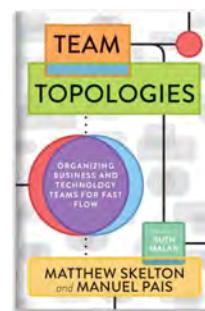
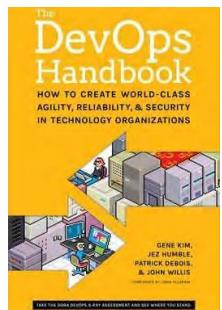
# The DevOps Collective Body of Knowledge

True to its core values, DevOps is emerging through a shared and collective body of knowledge (CBoK) including:

- Publications
- Conferences
- MeetUp groups
- Slack channels
- LinkedIn groups
- Videos and webinars
- Blogs and articles
- Case studies
- Awards
- Subject matter expertise

DevOps Institute actively researches and influences emerging DevOps practices in order to create meaningful training and certification.

# The DevOps Collective Body of Knowledge (2)

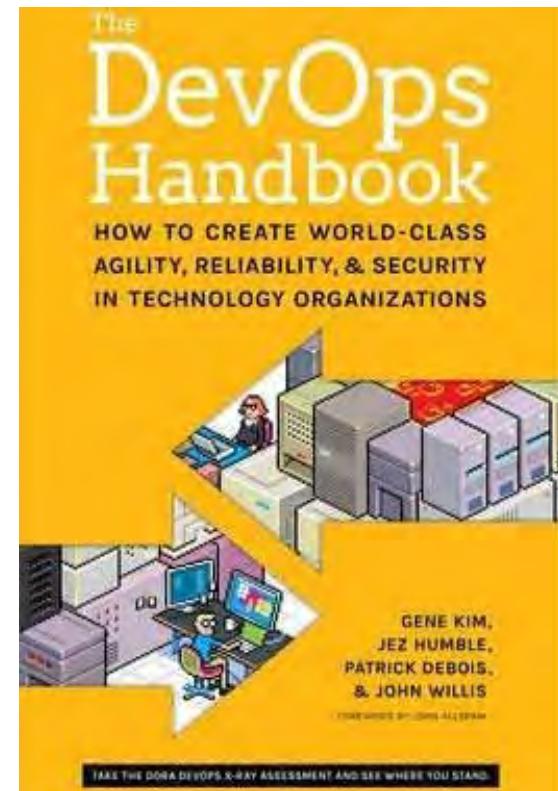




## The Short History of DevOps with Damon Edwards (11:47)

# 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."



"DevOps, in a sense, is about setting up a value delivery factory – a streamlined, waste-free pipeline through which value can be delivered to the business with a predictably fast cycle time."

Mark Schwartz  
*'The Art of Business Value'*



Module 1: Exploring DevOps

14

# What DevOps is NOT

- A title
- A separate team
- A tool
- Only culture
- Only automation
- Anarchy
- A one size fits all strategy



DevOps is coming to life through emerging practices that are delivering real value in real organizations.

# Why DevOps is Important Now

- Enterprises have young, nimble startup competitors
- Agile software development and cloud infrastructure is increasing
- IT can no longer operate in a silo culture
- More organizations are migrating to the cloud
- Consumers have “app” mentalities and expectations
- There is more data available to the business
- Time to value must accelerate

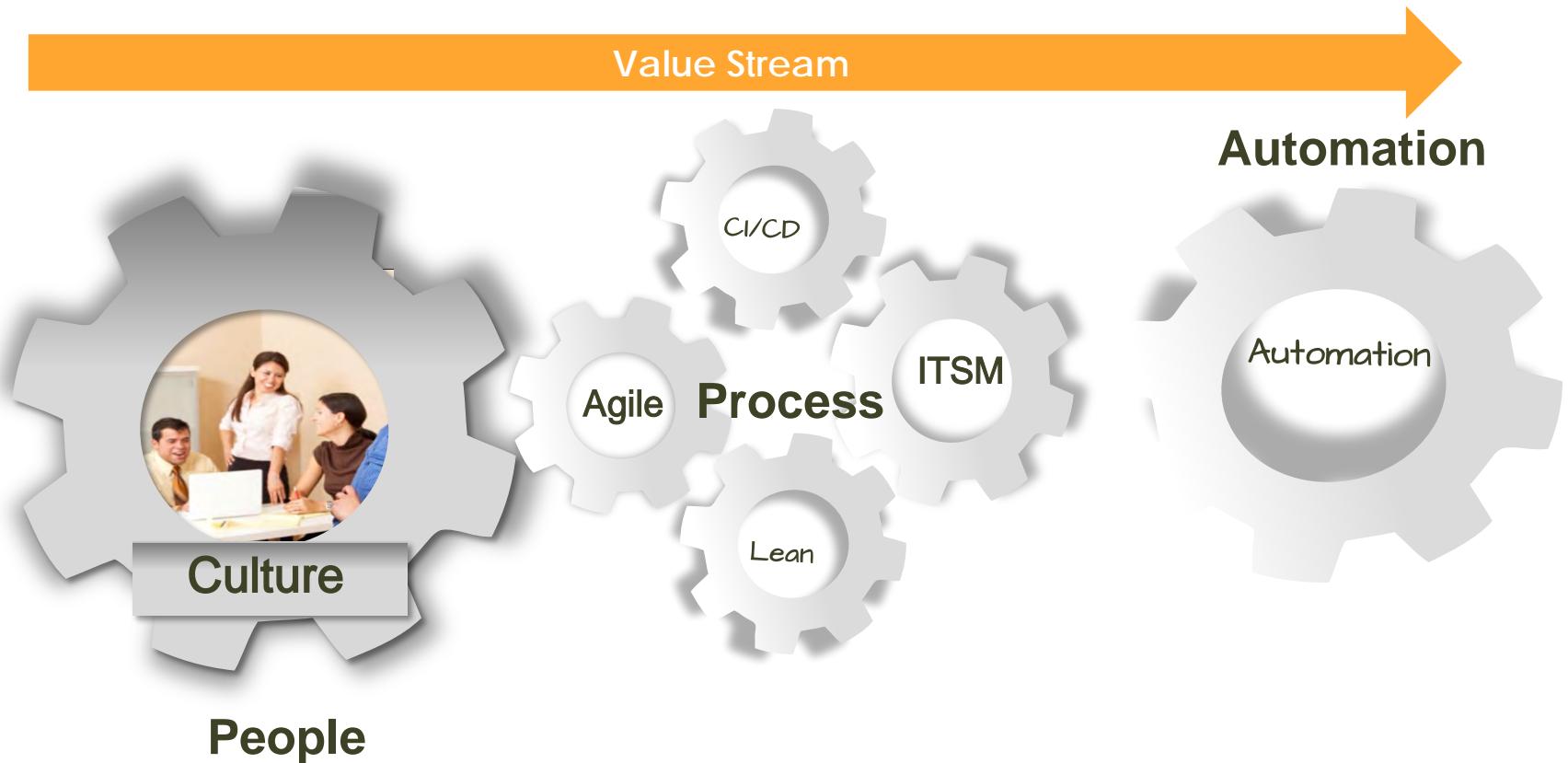
To meet these changing conditions, IT must adapt its culture, practices and automation to be more “continuous”.

# What Makes DevOps So Unique?

- Is it better than Scrum for improving the workflow of developers?
- Is it better than cybersecurity practices?
- Is it better than Lean in keeping IT more efficient?
- Is it better than ITIL® for service management?
- Is it better than Organizational Change Management for culture?
- Is it better than tools, technologies and automation?

Each of these frameworks and approaches have delivered some degree of benefit but none have delivered full end-to-end IT improvement.

# DevOps Applies Systems Thinking Across the Entire IT Spectrum

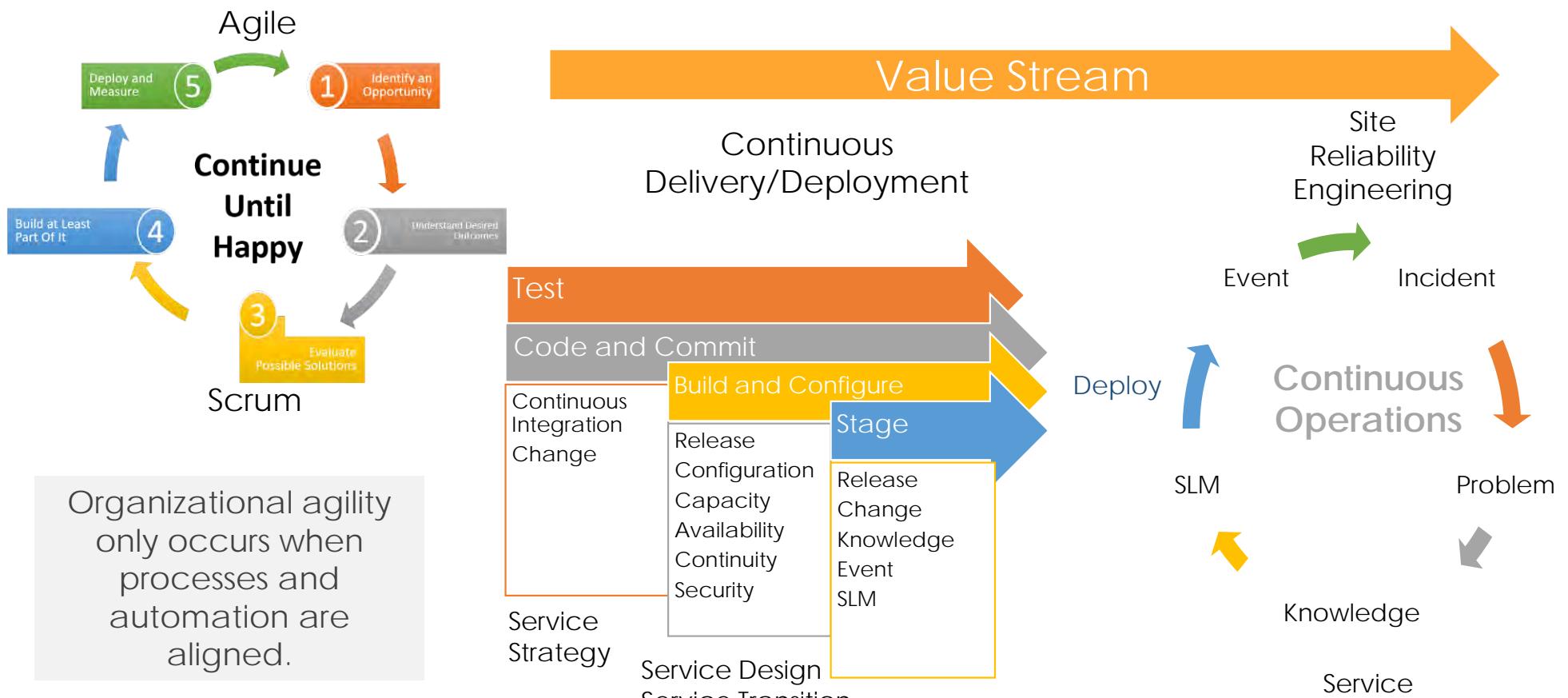


ITIL® is a registered trademark of AXELOS Limited.  
©DevOps Institute unless otherwise stated

Module 1: Exploring DevOps

18

# IT is a System of Systems



ITIL® is a registered trademark of AXELOS Limited.

©DevOps Institute unless otherwise stated

Module 1: Exploring DevOps

19

# DISCUSSION

## DevOps Myths versus Realities

# DevOps Goals

- Smaller, more frequent releases
- Reduced effort and risks
- Reduced cost of product iterations and delays
- A culture of communication and collaboration
- Consistency and speed through automation

## Improvements in

- Time to market/value
- Integration with the business
- Responsiveness
- Code and deployment quality
- Productivity
- Visibility
- Agility

# DevOps Values

More than anything else, DevOps is a cultural movement based on human and technical interactions to improve relationships and results.

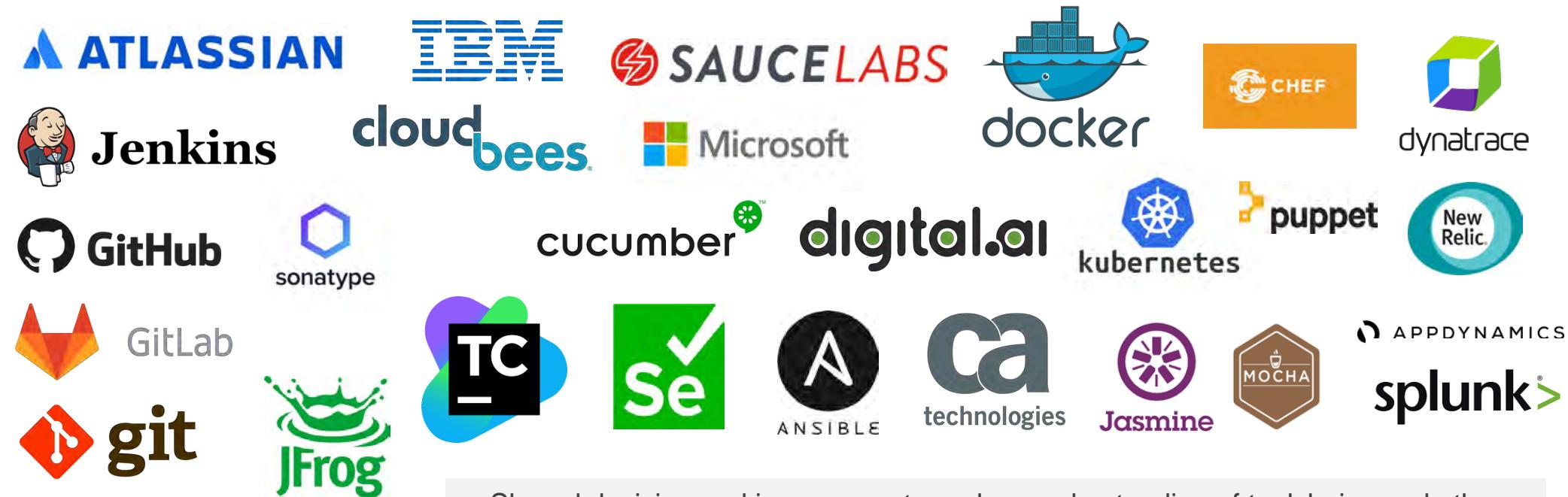


Module 1: Exploring DevOps

22

# Automation is an Essential Element

Automation enables agility, consistency, speed and reliability.



Shared decision-making, access to and an understanding of toolchains and other automation streamlines software delivery and prepares Ops for the long run.

Module 1: Exploring DevOps

©DevOps Institute unless otherwise stated

# DevOps Stakeholders



Dev includes all the people involved in developing software products and services including:

- Architects, business representatives, customers, product managers, project managers, quality assurance (QA) testers and analysts, suppliers, etc.

Ops includes all the people involved in delivering and managing software products and services including:

- Information security professionals, systems engineers, system administrators, IT operations engineers, release engineers, database administrators (DBAs), network engineers, support professionals, suppliers, etc.

DevOps extends beyond software developers and IT operations.





“You never change things by fighting the existing reality. To change something, build a new model which makes the existing model obsolete.”

Buckminster Fuller

Module 1: Exploring DevOps

# Why Does DevOps Matter?

# Our Cadence is Off

Historically...



Cadence – the flow or rhythm of events.

# DevOps Improves IT's Cadence and Velocity

...agile, lean and ITSM practices are also needed.

## The Business



Winning through  
Innovation

## Agile/Lean DevOps



Continuous Delivery



# DevOps Improves Throughput AND Stability

According to the 2019 State of DevOps Report, elite-performing organizations have:

- **208** times more frequent code deployments
- **106** times faster lead time from commit to deploy
- **2604** times faster time to recover from incidents
- **7** times lower change failure rate

"Our research continues to show that the industry-standard Four Key Metrics of software development and delivery drive organizational performance in technology transformations.

This year's report revalidates previous findings that it is possible to optimize for stability without sacrificing speed."

"The Accelerate State of DevOps Report represents six years of research and data from over 31,000 professionals worldwide. It is the largest and longest-running research of its kind, providing an independent view into the practices and capabilities that drive high performance. The results let us understand the practices that lead to excellence in technology delivery and powerful business outcomes."



Module 1: Exploring DevOps

29

# CASE STORY: ING Bank

"We wanted to establish a culture and environment where building, testing and releasing software can happen rapidly, frequently and more reliably. When beginning this journey we started with what matters most: people. There was a beginning to this journey, but there will be no end. An end would put a stop to the transformation, while in fact you always need to make sure you keep getting better for the customer."



Ron van Kemenade,  
CIO

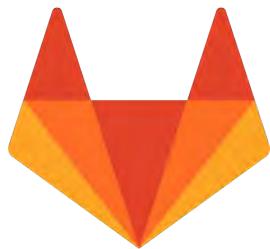
## Benefits

- Transformed from risk averse organization to agile powerhouse
- Improved time to market from 13 weeks to less than 1 week
- More automated processes
- A sharp reduction of handovers
- A collaborative performance culture



"IT has  
become the  
beating heart  
of the bank."

# DevOps Adoption



GitLab 2020  
DevSecOps  
Survey

- Nearly 60% deploy multiple times a day, once a day, or once every few days. Up from 45% in 2019
- Over 25% of companies are in the DevOps “sweet spot” of three to five years of practice
- 47% of companies say testing is the number one reason for delays



2020 State  
of DevOps  
Report

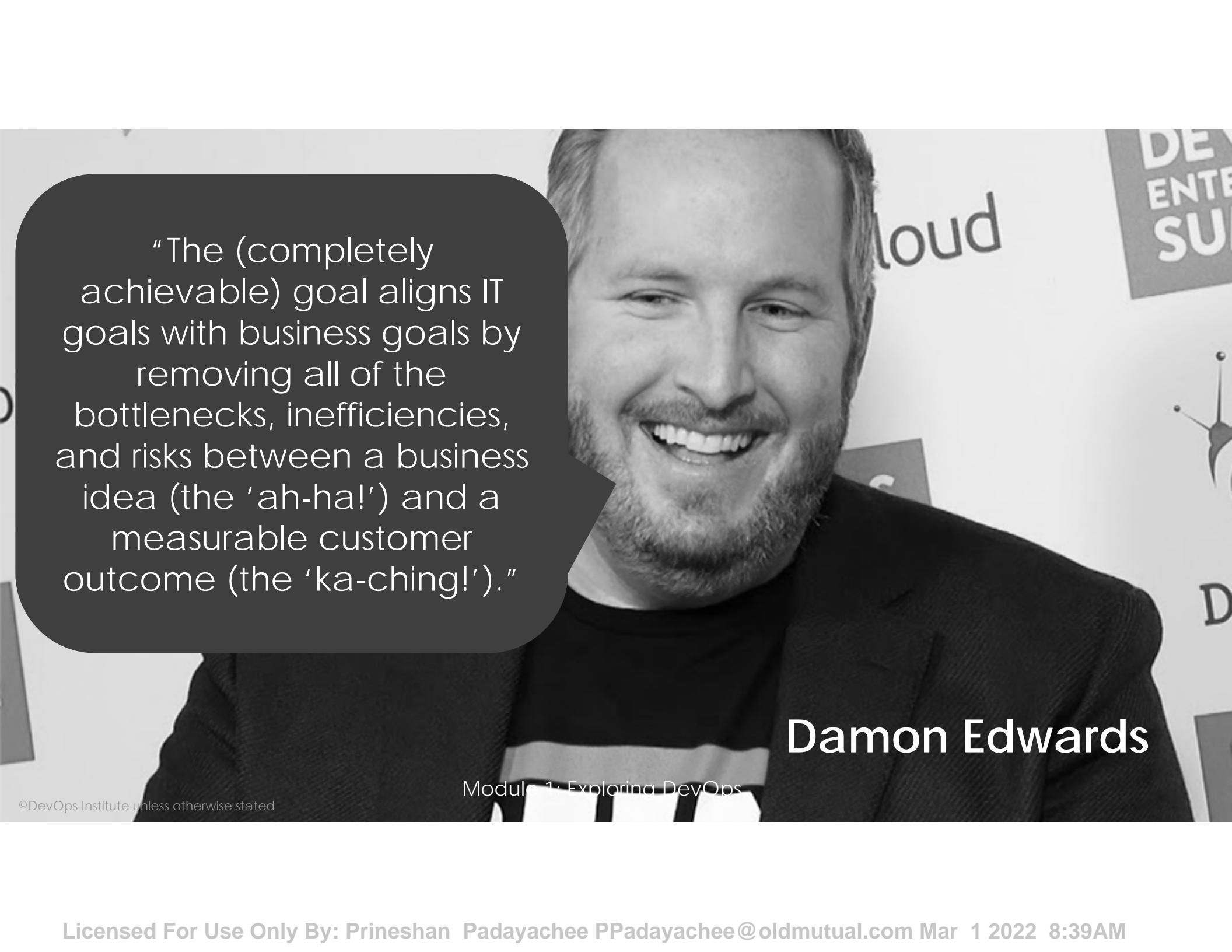
- Highly evolved firms are six times as likely to report high use of internal platforms as firms at a low level of DevOps evolution
- Highly evolved firms are nearly three times as likely to have highly effective change management as firms at a low level of DevOps evolution



21  
Predictions  
for 2021

- Business leaders will increasingly value DevOps
- Hybrid product teams will become a pillar of customer value delivery
- Application security will no longer be an afterthought
- Predictive DevOps will be the next transformation that will deliver business value

Module 1: Exploring DevOps



"The (completely achievable) goal aligns IT goals with business goals by removing all of the bottlenecks, inefficiencies, and risks between a business idea (the 'ah-ha!') and a measurable customer outcome (the 'ka-ching!')."

Damon Edwards

Module 1: Exploring DevOps

©DevOps Institute unless otherwise stated

# The Business Perspective

# Why the Business is Driving DevOps

- Every business has become a tech business
- IoT is rapidly increasing
- Consumers have developed “app” mentalities
- Customers value outcomes, not products
- Time to value is replacing time to market
- Intelligent data must shape direction quickly
- Customer delight is more important than customer satisfaction

Your biggest competitor may be a start-up.

# The Business Value of DevOps

Combined, commercial and non-commercial goals include:

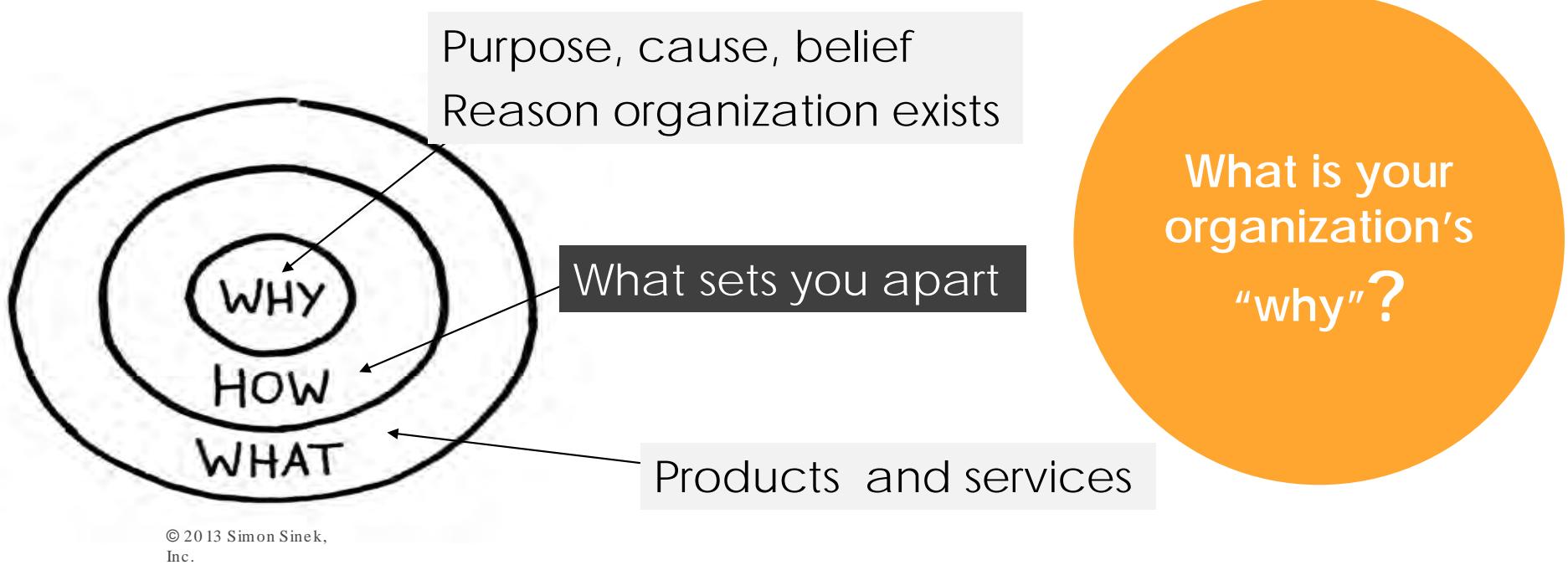
- Profitability
- Productivity
- Market share
- Number of customers
- Quantity of products or services
- Operating efficiency
- Customer satisfaction
- Quality of products or services provided
- Achieving organization or mission goals



**"Delivering software quickly, reliably, and safely is at the heart of technology transformation and organizational performance."**

We see continued evidence that software speed, stability, and availability contribute to organizational performance (including profitability, productivity, and customer satisfaction). Our highest performers are twice as likely to meet or exceed their organizational performance goals."

# Start with the “Why” - The Golden Circle





Start With Why  
with Simon Sinek (5:00)



# EXERCISE

## Your Organizational Why



"Agile was instrumental in Development regaining the trust in the business, but it unintentionally left IT Operations behind. DevOps is a way for the business to regain trust in the entire IT organization as a whole."

**Clyde Logue**  
**Founder of StreamStep**

# The IT Perspective

Module 1: Exploring DevOps

©DevOps Institute unless otherwise stated

# Why IT is Driving DevOps

- Every business has become a tech business
- IoT is rapidly increasing
- Consumers have developed “app” mentalities
- Customers value outcomes, not products
- Time to value is replacing time to market
- Intelligent data must shape direction quickly
- Customer delight is more important than customer satisfaction

Do you  
recognize  
these drivers?

Statistics

Module 1: Exploring DevOps

41

POWER C  
POWER C  
EMAIL  
INTERNET  
WORLD  
NO  
SEARCH

# The IT Challenge

DevOps must continuously deliver outcomes by bridging and improving almost every aspect of IT.

## Internal IT challenges:

- IT must go faster, faster, faster without risking quality
- Prior investments aren't delivering end to end value
  - Agile SW development is good but isn't delivering full value
  - ITSM processes are good but aren't delivering full value
  - New automation is good but isn't delivering full value
- IT's silo culture is constraining the value stream

IT no longer needs to align or integrate with the business,  
IT is the business.

# The Wall of Confusion (1)

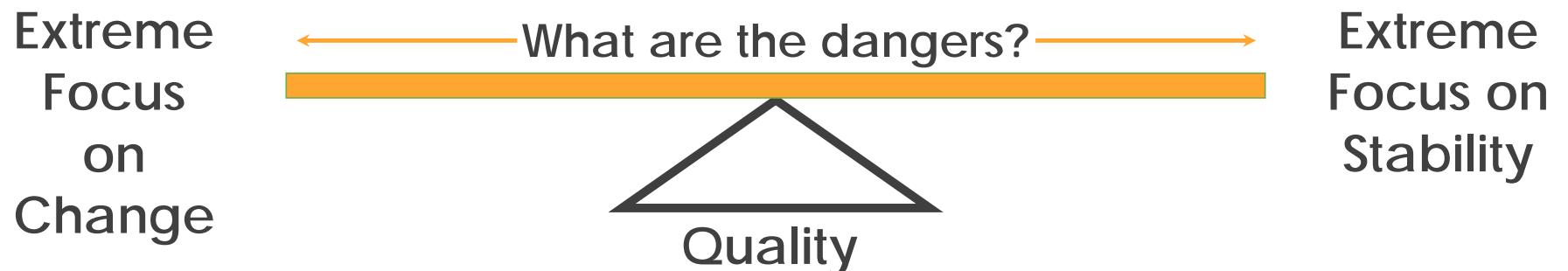
What about  
Security,  
Governance, Risk  
Management and  
Compliance?  
What do they  
want?



Module 1: Exploring DevOps

43

# The Wall of Confusion (2)



What does the business want?  
All of the above

# IT's Silo Culture

Dev



Ops



QA  
Security

Service Desk

Isolated IT silos can foster stereotypes and misconceptions between Dev, Ops and other IT teams.

It can also affect the flow of work and IT's ability to deliver innovation continuously.

# Module One Quiz

- 1 What does CALMS stand for?
  - a) Culture, Automation, Lean, Management, Sharing
  - b) Collaboration, Automation, Lean, Metrics, Sharing
  - c) Culture, Automation, Lean, Measurement, Sharing
  - d) Continuous Integration, Automation, Lean, Measurement, Sharing
- 2 Who first coined the word 'DevOps'?
  - a) Gene Kim
  - b) Patrick Debois
  - c) John Willis
  - d) Damon Edwards
- 3 Who became the partner with DORA for the Accelerate State of DevOps Reports in 2018?
  - a) Puppet
  - b) Chef
  - c) Google
  - d) Amazon Web Services
- 4 In 2015, Gartner predicted what percentage of Global 2000 organizations would have DevOps as a mainstream strategy?
  - a) 100%
  - b) 50%
  - c) 20%
  - d) 10%
- 5 Who is responsible for The Golden Circle (organizational why)?
  - a) Simon Sinek
  - b) Alan Alter
  - c) Ron van Kemenade
  - d) Jez Humble

# Module One Quiz Answers

- |   |   |
|---|---|
| 1 What does CALMS stand for?  | a) Culture, Automation, Lean, Management, Sharing<br>b) Collaboration, Automation, Lean, Metrics, Sharing<br><b>c) Culture, Automation, Lean, Measurement, Sharing</b><br>d) Continuous Integration, Automation, Lean, Measurement, Sharing |
| 2 Who first coined the word 'DevOps'?   | a) Gene Kim<br><b>b) Patrick Debois</b><br>c) John Willis<br>d) Damon Edwards   |
| 3 Who became the partner with DORA for the Accelerate State of DevOps Reports in 2018?                                | a) Puppet<br>b) Chef<br><b>c) Google</b><br>d) Amazon Web Services  |
| 4 In 2015, Gartner predicted what percentage of Global 2000 organizations would have DevOps as a mainstream strategy? | a) 100%<br>b) 50%<br><b>c) 20%</b><br>d) 10%  |
| 5 Who is responsible for The Golden Circle (organizational why)?  | <b>a) Simon Sinek</b><br>b) Alan Alter<br>c) Ron van Kemenade<br>d) Jez Humble  |

# Module 2

## CORE DEVOPS PRINCIPLES

# Module 2: Core DevOps Principles

- The Three Ways
- The Theory of Constraints
- Chaos Engineering
- Learning Organizations

Component	Module 2 Content
Video	Gene Kim Defines the Three Ways of The Phoenix Project
Case Story	Ticketmaster
Discussion	Overcoming Constraints
Exercise	Bringing The Three Ways to Life

# The Three Ways

Module 2: Core DevOps Principles

©DevOps Institute unless otherwise stated

# The Three Ways



The First Way	The Second Way	The Third Way
Flow	Feedback	Continuous Experimentation & Learning
Understand and increase the flow of work (left to right)	Create short feedback loops that enable continuous improvement (right to left)	<p>Create a culture that fosters:</p> <ul style="list-style-type: none"><li>• Experimentation, taking risks and learning from failure</li><li>• Understanding that repetition and practice is the prerequisite to mastery</li></ul>

## Module 2: Core DevOps Principles



Gene Kim Defines The Three Ways  
of The Phoenix Project (3:31)

# The First Way: Flow



- Understanding the flow of work
- Increasing flow by understanding and removing constraints
- Never passing a known defect downstream
- Never allowing local optimization to cause global degradation
- Achieving a profound understanding of the entire system

A goal of The First Way is to have work flow quickly from left to right.

# Theory of Constraints

A methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.

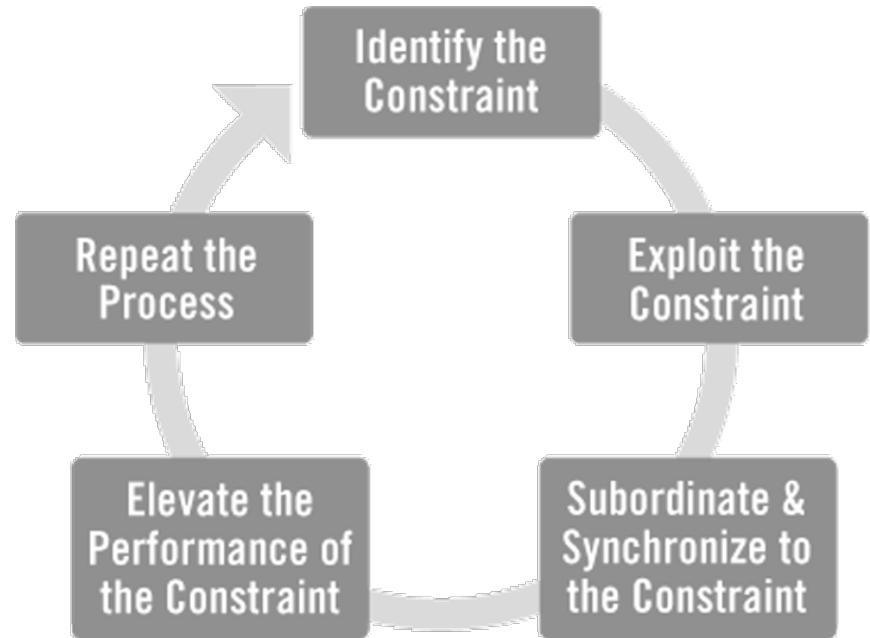
The Theory of Constraints recognizes that

- Every process has at least one constraint or bottleneck that affects its ability to consistently meet its goal
- The process will only meet the capacity of its constraints and will be only as successful as its weakest link
- Improving constraints is the fastest and most efficient way to improve the entire process or system

The Theory of Constraints was introduced in the book  
'The Goal' by Eliyahu M. Goldratt.

# Common Constraints

- Development delays
- Environment creation (test, staging, production, etc.)
- Code deployment
- Test setup and run
- Security or QA assessments
- Overly tight architecture
- Product management
- Complex or bureaucratic processes



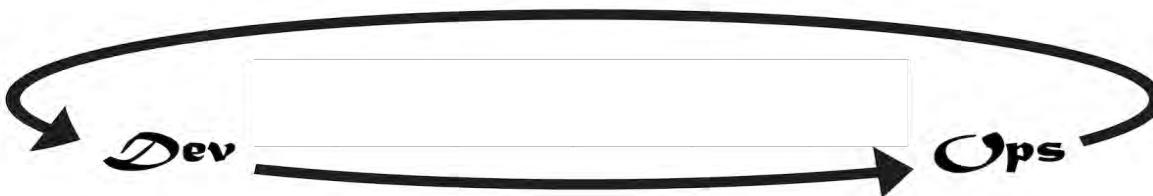
# DISCUSSION

## Overcoming Constraints

Module 2: Core DevOps Principles

©DevOps Institute unless otherwise stated

# The Second Way: Feedback



- Understand and respond to the needs of all customers – both internal and external
- Shorten and amplify all feedback loops
- Create and embed knowledge where needed

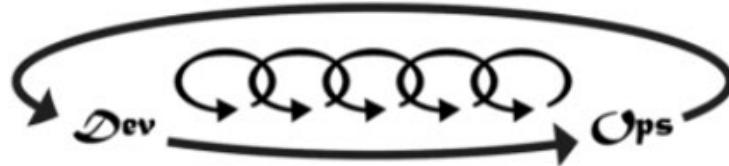
A goal of The Second Way is to shorten and amplify right to left feedback loops so necessary corrections can be continually made.

# Examples of Feedback Loops



- Automated testing
  - Peer review of production changes
  - Monitoring/Event Management data
  - Dashboards
  - Production logs
- Process measurements
  - Post-mortems
  - Shared on-call rotation
  - Change, Incident, Problem and Knowledge Management data

## The Third Way: Continual Experimentation and Learning



The Third Way encourages a culture that fosters two things:

1. Continual experimentation, taking risks and learning from failure
2. Understanding that repetition and practice is the prerequisite to mastery.

- Allocate time for the improvement of daily work
- Create rituals that reward the team for taking risks
- Introduce faults into the system to increase resilience
- Plan time for safe experimentation and innovation (hackathons)

# Chaos Engineering

- The 'Simian Army' concept was first adopted by Netflix as a service that randomly terminates a production instance
- Response to attacks helps to build competencies to recover the production environment from inevitable failures



"Chaos Monkey is a tool that randomly disables our production instances to make sure we can survive this common type of failure without any customer impact. The name comes from the idea of unleashing a wild monkey with a weapon in your data center (or cloud region) to randomly shoot down instances and chew through cables - all the while we continue serving our customers without interruption. By running Chaos Monkey in the middle of a business day, in a carefully monitored environment with engineers standing by to address any problems, we can still learn the lessons about the weaknesses of our system, and build automatic recovery mechanisms to deal with them. So next time an instance fails at 3 am on a Sunday, we won't even notice." Netflix

# CASE STORY: Ticketmaster

"Over the years, we have been extracting these legacy technologies by adding APIs to modernize the interface to our ticketing engines and platforms. We wanted to get them out quickly. To do that, we need to touch a lot of systems.

This has driven us to DevOps. For us, it really started with DevOps. Part of our transformation was to focus on delivering business value faster and delivering more of it, and the driver was speed to market of product."

"There is less emphasis on the amount of work being done, and more on the outcome."



Justin Dean, VP  
TechOps

## Benefits

- Removal of legacy bottlenecks and constraints
- Improvement of speed to market
- Outcome and business value focused
- Authority distributed, greater autonomy
- Friction reduced through self-service

Module 2: Core DevOps Principles

61

©DevOps Institute unless otherwise stated



# Encourage a Learning Culture

- Encourage daily learning and knowledge sharing
- Create training and skills-based education plans
- Incorporate learning into processes
- Use technology to accelerate learning
- Make work educational through experimentation, problem solving and demonstrations
- Allow and use mistakes as sources of learning
- Make the results of learning visible

"You're either a learning organization or you're losing to somebody who is."

Andrew Shafer quoted in  
'Beyond the Phoenix Project'

## EXERCISE

# Bringing the Three Ways to Life

Module 2: Core DevOps Principles

©DevOps Institute unless otherwise stated

# Module Two Quiz

- 1 Which of The Three Ways is concerned with feedback loops like the results of automated tests?
  - a) The First Way
  - b) The Second Way
  - c) The Third Way
  - d) All of The Three Ways
- 2 How do DevOps principles frame the treatment of failure?
  - a) As a learning opportunity
  - b) As something to be ignored
  - c) That it should be punished
  - d) That we can't protect against it
- 3 What is the prerequisite to mastery?
  - a) Talent & capability
  - b) Training & study
  - c) Guidance & mentoring
  - d) Repetition & practice
- 4 We should:
  - a) Never pass a known defect downstream
  - b) Seek an overview of the system
  - c) Allow local optimization to degrade global optimization
  - d) Optimize all links in a process
- 5 Which organization created the Chaos Monkey and made it open source?
  - a) Facebook
  - b) Etsy
  - c) Google
  - d) Netflix

# Module Two Quiz Answers

- 1 Which of The Three Ways is concerned with feedback loops like the results of automated tests?
  - a) The First Way
  - b) The Second Way**
  - c) The Third Way
  - d) All of The Three Ways
- 2 How do DevOps principles frame the treatment of failure?
  - a) As a learning opportunity**
  - b) As something to be ignored
  - c) That it should be punished
  - d) That we can't protect against it
- 3 What is the prerequisite to mastery?
  - a) Talent & capability
  - b) Training & study
  - c) Guidance & mentoring
  - d) Repetition & practice**
- 4 We should:
  - a) Never pass a known defect downstream**
  - b) Seek an overview of the system
  - c) Allow local optimization to degrade global optimization
  - d) Optimize all links in a process
- 5 Which organization created the Chaos Monkey and made it open source?
  - a) Facebook
  - b) Etsy
  - c) Google
  - d) Netflix**

# Module 3

## KEY DEVOPS PRACTICES

# Module 3: Key DevOps Practices

- Continuous:
  - Testing
  - Integration, Delivery, Deployment
- Site Reliability & Resilience Engineering
- DevSecOps
- ChatOps
- Kanban

Component	Module 3 Content
Video	GitHub Professional Guides: Continuous Integration & Delivery
Case Story	Capital One
Discussion	Why Too Much WIP is Bad
Exercise	Rate Your CI/CD Capability

# CASE STORY: Capital One

"Tools are a big part of today's Agile and DevOps methodologies. A typical project deals with Agile Project Management tools, Source Control, Continuous Integration (CI) tool, Testing tools, Static Code Analysis and Security Scanning tools, Deployment and Monitoring tools to name a few. Large enterprises and complex systems sometimes use multiple CI, Testing and Scanning tools. Each of these has nice dashboards to present key information stored in it. But what is lacking is a single, comprehensive end-to-end view of the state of a delivery pipeline in near real time. At Capital One, we believe that while tools, automation and collaboration are very important, a continuous feedback loop is critical to DevOps success."

"Driven by data, technology, and data science."



Topo Pal,  
Director & Platform  
Engineering Fellow

## Benefits

- 100s of code commits per day
- Integration from once a month to every 15 minutes
- QA from once per month to 4 times per day
- Deployment from manual to completely automated
- Production release from monthly/quarterly to once per sprint

## Module 3: Key DevOps Practices

68

# Continuous:

- Testing
- Integration
- Delivery
- Deployment

# Continuous Testing

Continuous testing is the process of executing automated tests as part of the deployment pipeline to obtain immediate feedback on the business risks associated with a software release candidate.



	Low	Medium	High	Elite
Automated build	64%	81%	91%	92%
Automated unit tests	57%	66%	84%	87%
Automated acceptance tests	28%	38%	48%	58%
Automated performance tests	18%	23%	18%	28%
Automated security tests	15%	28%	25%	31%
Automated provisioning and deployment to testing environments	39%	54%	68%	72%

“Shifting left” is about building quality into the software development process. When you shift left, fewer things break in production, because any issues are detected and resolved earlier.

# Continuous Integration

Continuous integration (CI) is a development practice that requires developers to commit code into a shared repository (master/trunk) at least daily.

- 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

While mostly associated with agile software development, waterfall approaches can also take advantage of continuous integration and test-driven development practices.

# CI/CD Toolchain Skills Are Essential for DevOps

"In short, CI/CD toolchains help with velocity and quality of code, allow for better collaboration among the teams and automates many steps, tasks and processes which reduced the risk and cost of software development."

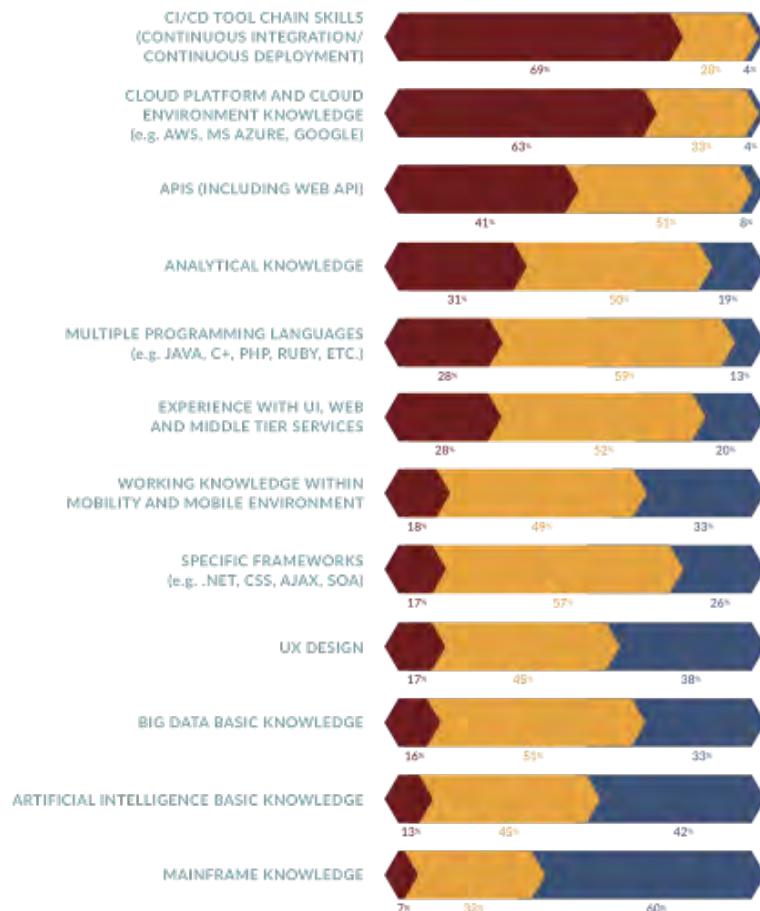
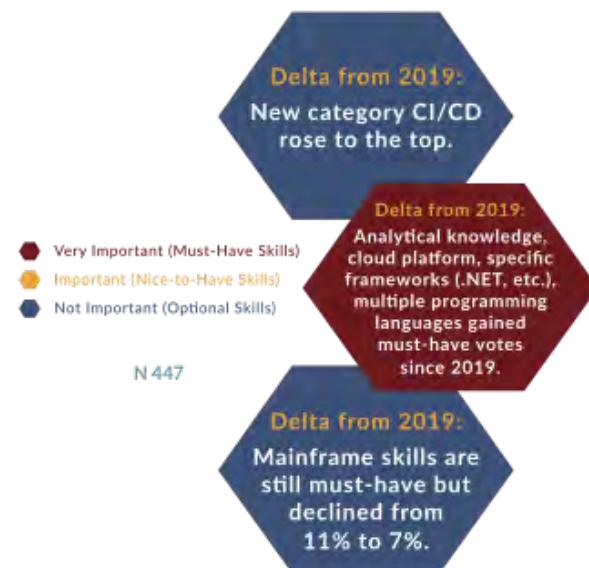


Figure 13: Technical Skills For The DevOps Human CI/CD Toolchain, Cloud Platform And Understanding APIs Are The Top 3 Must-have Technical Skills

Q How would you rate the importance of the following technical skills for your DevOps team members?



# Continuous Delivery

Continuous delivery is a methodology that focuses on making sure software is *always in a releasable state* throughout its lifecycle.

- 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

**Factors that positively contribute to continuous delivery:**



Module 3: Key DevOps Practices

73

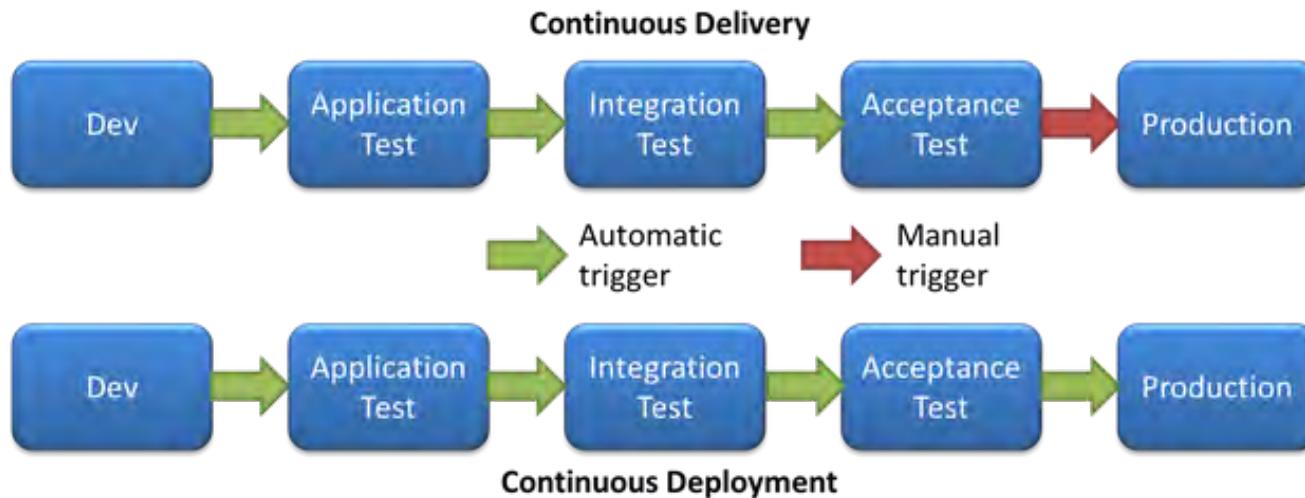
©DevOps Institute unless otherwise stated

## Module 3: Key DevOps Practices

# EXERCISE

# Rate Your CI/CD Capability

# Continuous Delivery & Continuous Deployment



Continuous integration is the practice that allows for the principle of continuous delivery of value into users' hands.

From: Mirco Hering: [notafactoryanymore.com](http://notafactoryanymore.com),  
author of 'DevOps for the Modern Enterprise'

Module 3: Key DevOps Practices

75



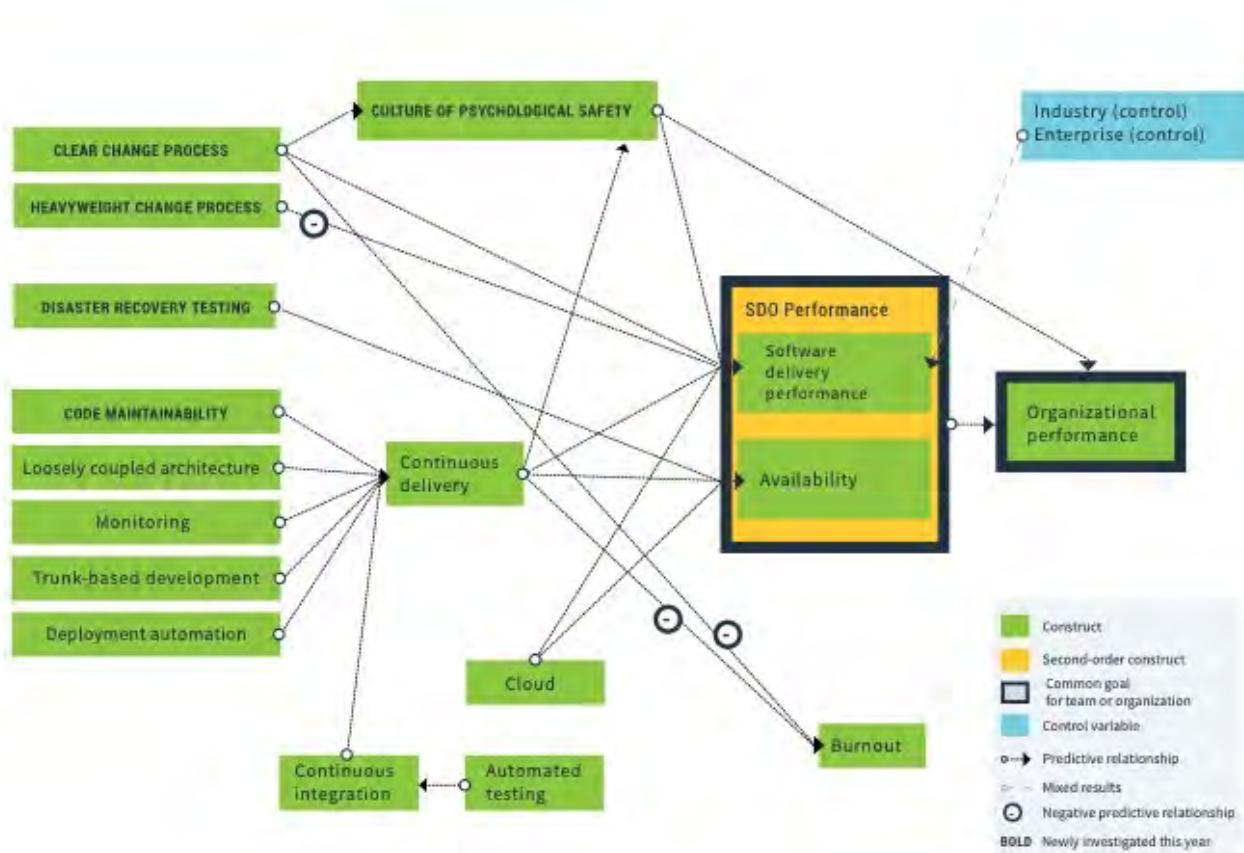
# GitHub

## Professional Guides



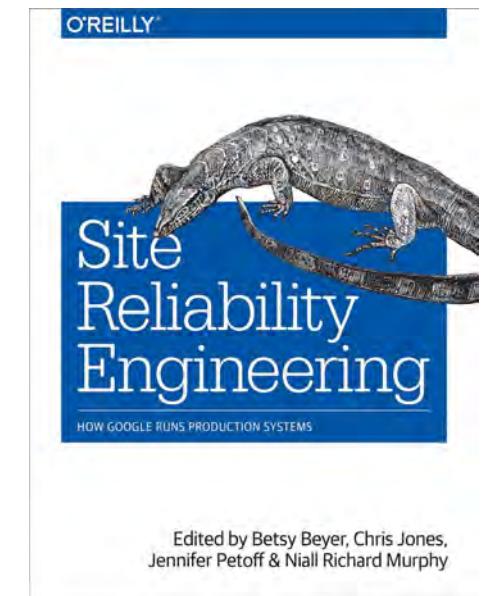
GitHub Professional Guide:  
Continuous Integration & Delivery (6:00)

# Continuous Delivery Leads to Higher Organizational Performance



# Site Reliability Engineering

- "What happens when a software engineer is tasked with what used to be called operations." Ben Treynor, Google
- Goals are to create ultra-scalable and highly reliable software systems
- 50% of their time doing "ops" related work such as issues, on-call, and manual intervention
- 50% of their time on development tasks such as new features, scaling or automation



Google now has over 1,500 Site Reliability Engineers

# Resilience Engineering

The intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions.

- Resilience engineering looks at how the organization functions as a whole
- The best defense is a good offense
- Take an aggressive, blameless and systemic view post incident
- Consider both human and technical elements
- Systems must be stronger than their weakest link

"Failure is the flip side of success." Eric Hollnagel

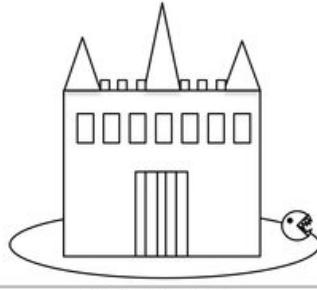
# DevSecOps

The purpose and intent of DevSecOps is to build on the mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

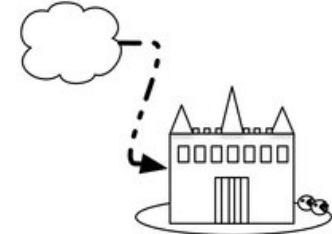
[www.devsecops.org](http://www.devsecops.org)

## Who turned out the lights?

Sire, all is good with our layered defenses... Our moat keeps out the unauthorized thieves and the castle protects our most precious assets. We have secret rooms and passages that make it impossible to steal from us. We are secure.



Sire, our plan is to put the "cloud" inside our castle to protect our assets... We'll need a second alligator to ensure we have coverage. We know how to operate this way and there is nothing different about the "cloud"... We assure you we've got it...



(c) 2015 devsecops.org



- Introduces security as code
- Embraces the "shift left" testing strategy
- Leverages automation for resilience, testing, detection and audit
- Breaks the security constraint

## Module 3: Key DevOps Practices

80

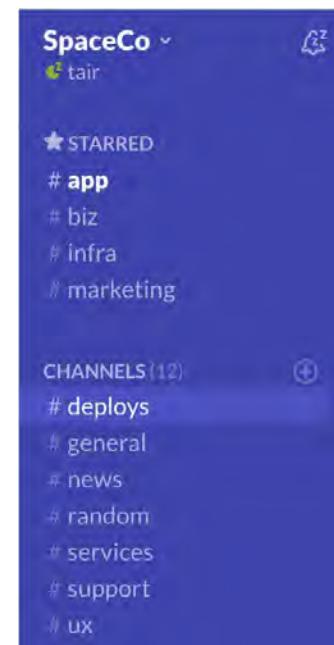
© DevOps Institute unless otherwise stated

# ChatOps

Group chat client + chat bots = conversation-driven development, delivery and support



Google Hangouts



3 members Shipping teleport to Mars

ahmed 10:30 PM so @tair, how about deploying that teleport fe

tair 10:32 PM let's ☕

Dockbit BOT 10:34 PM Deployment of RocketApp by @tair

#1014: rocket-app/master Deployment scheduled. Click to see the pra

tair 10:35 PM I'll grab some ☕ while it's deploying...

Dockbit BOT 10:35 PM Deployment of RocketApp by @tair

#1014: rocket-app/master Deployment completed successfully.

The transparency of ChatOps shortens feedback loops, improves information sharing, enhances team collaboration and enables cross-training. It can also be used to decrease MTTR.

# Kanban

Kanban is a method of work that pulls the flow of work through a process at a manageable pace.

- Visualizes and manages workflow
- Pulls work for teams when they are ready for it
- Enables people to work collaboratively to improve flow
- Measures team velocity (quantity of work done in an iteration)
- Reduces idle time and waste in a process



- Makes work visible
- Makes policies explicit
- Limits work in progress (WIP) to capacity

## DISCUSSION

### Why Too Much WIP is Bad

Module 3: Key DevOps Practices

©DevOps Institute unless otherwise stated

# Module 3: Quiz

- 1 Which of the following is not needed for Continuous Integration?
  - a) Developers commit code to trunk/master at least daily
  - b) Push button deployment
  - c) Unit, integration and user acceptance tests
  - d) Consistent coding standards
- 2 Which of the following is not a non-functional test?
  - a) Performance
  - b) Unit
  - c) Security
  - d) Capacity
- 3 How many Site Reliability Engineers does Google have?
  - a) 15
  - b) 150
  - c) 1,500
  - d) 15,000
- 4 Which of these is not a ChatOps platform?
  - a) Jira
  - b) Slack
  - c) Stride
  - d) Teams
- 5 Which of these is not true about Kanban?
  - a) Makes work visible
  - b) Pushes work through
  - c) Applies WIP limits
  - d) Measures team velocity

84

# Module 3: Quiz Answers

- 1 Which of the following is not needed for Continuous Integration?
  - a) Developers commit code to trunk/master at least daily
  - b) Push button deployment**
  - c) Unit, integration and user acceptance tests
  - d) Consistent coding standards
- 2 Which of the following is not a non-functional test?
  - a) Performance
  - b) Unit**
  - c) Security
  - d) Capacity
- 3 How many Site Reliability Engineers does Google have?
  - a) 15
  - b) 150
  - c) 1,500**
  - d) 15,000
- 4 Which of these is not a ChatOps platform?
  - a) Jira**
  - b) Slack
  - c) Stride
  - d) Teams
- 5 Which of these is not true about Kanban?
  - a) Makes work visible
  - b) Pushes work through**
  - c) Applies WIP limits
  - d) Measures team velocity

# Module 4

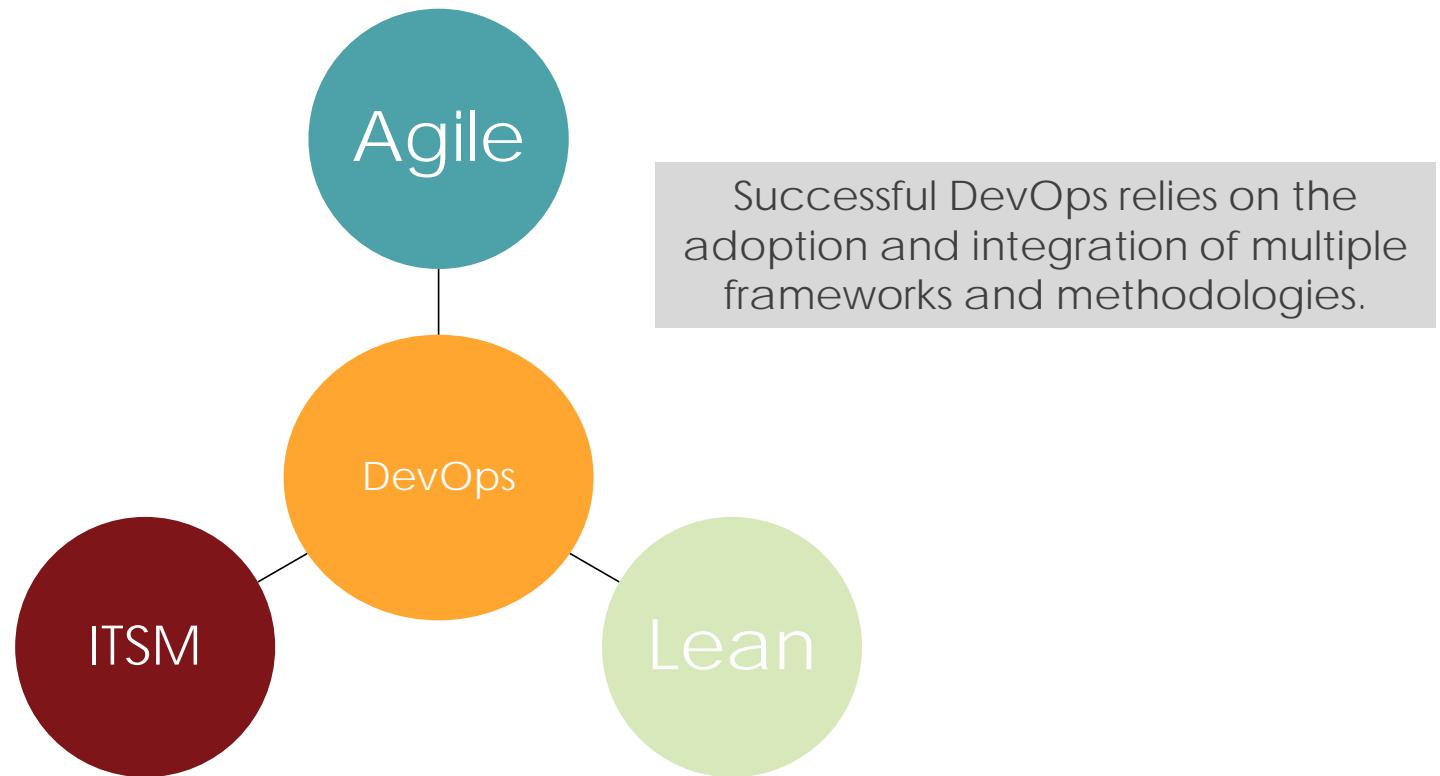
# BUSINESS & TECHNOLOGY FRAMEWORKS

# Module 4: Business & Technology Frameworks

- Agile
- ITSM
- Lean
- Safety Culture
- Learning Organizations
- Continuous Funding

Component	Module 4 Content
Video	Spotify Engineering Culture Part 1
Case Story	Alaska Air
Discussion	Agility in IT Operations
Exercise	Identifying & Eradicating Waste

# DevOps Cannot Stand Alone



Module 4: Business & Technology Frameworks

88

©DevOps Institute unless otherwise stated

# Agile

Module 4: Business & Technology Frameworks

©DevOps Institute unless otherwise stated

# The Agile Manifesto

The underlying concepts of agile software development were first laid out in the Agile Manifesto.



It is more important to BE agile than DO Agile:

- Be customer-centric
- Be lean
- Be collaborative
- Be communicative
- Be adaptive
- Be measurable
- Be consistent
- Be results-oriented
- Be reflective

# Scrum

Scrum is a **simple** framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "**just enough**" structure for teams to be able to focus their **innovation** on solving what might otherwise be an insurmountable challenge.

Scrum.org

Scrum is

- The most commonly applied Agile software development practice
- Deceptively simple yet difficult to master
- Not a process or a technique for building *products*

Scrum increases the ability to release more frequently.

<http://www.scrumguides.org/>

©DevOps Institute unless otherwise stated

Module 4: Business & Technology Frameworks



## Roles

- Product Owner
- ScrumMaster
- Development Team



## Artifacts

- Product Backlog
- Sprint Backlog
- Increment

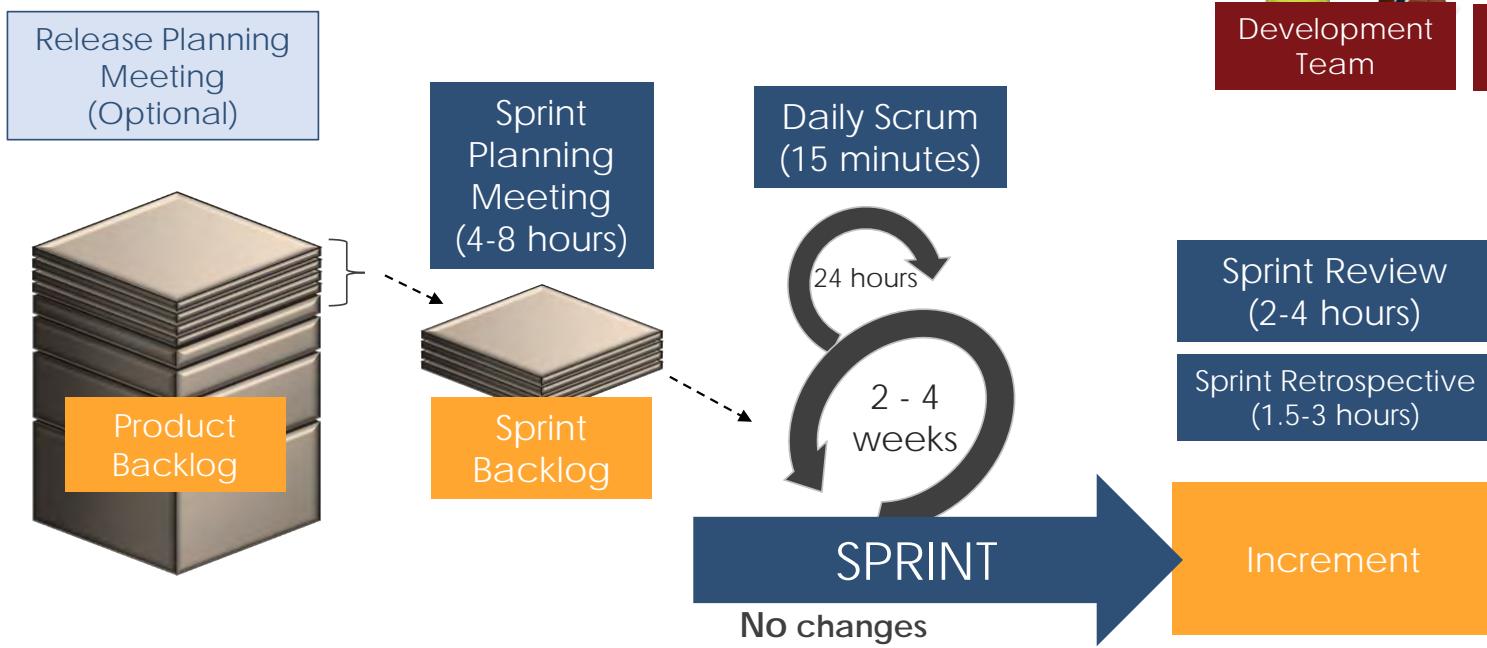


## Meetings

- The Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

# Scrum in a Nutshell

Scrum = 3 Roles + 3 Artifacts + 5 Events



Where do IT Operations get involved?

Module 4: Business & Technology Frameworks

92

# The Product Backlog

- **Stories**, also called “user stories,” are short requirements or requests written from the perspective of an end user.
- **Epics** are large bodies of work that can be broken down into a number of smaller tasks (called stories).
- **Initiatives** are collections of epics that drive toward a common goal.
- **Themes** are large focus areas that span the organization.



# Scaled Agile Framework® (SAFe™)

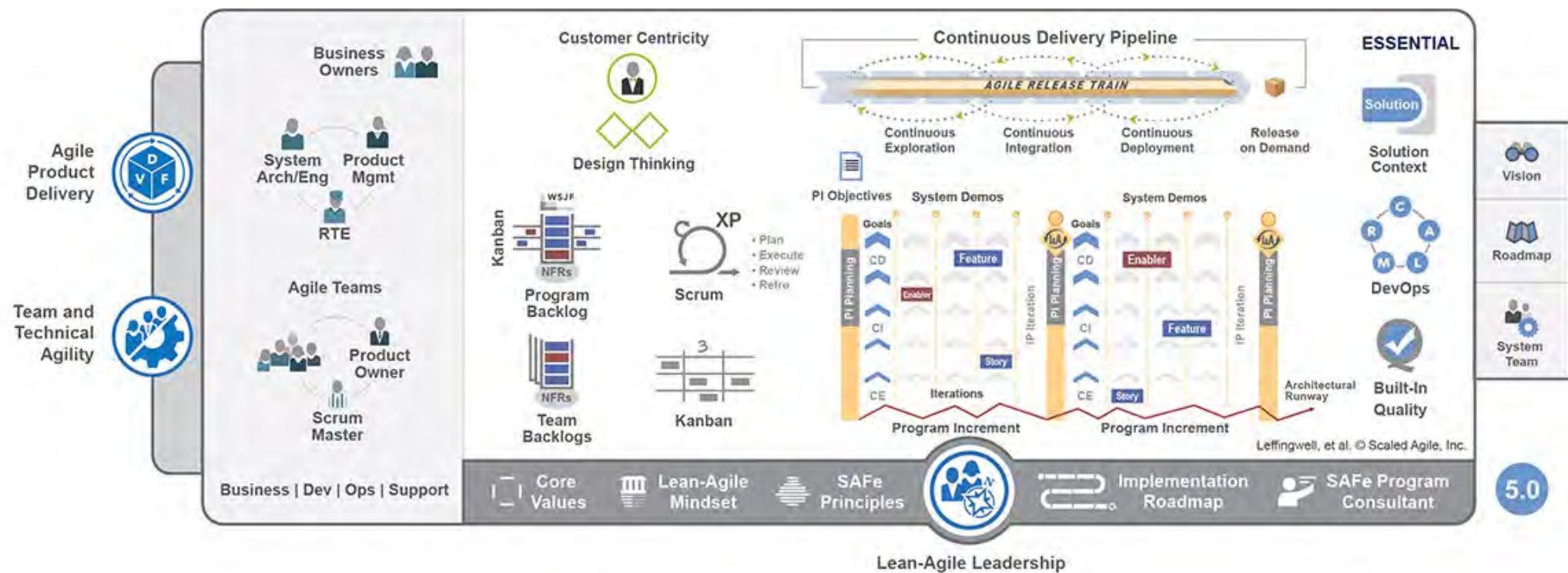
The Scaled Agile Framework (SAFe) is a proven, publicly available framework for applying Lean-Agile principles and practices at enterprise scale.

- Integrates Lean and Agile thinking into software development
- Focuses on iterative and incremental development, agile SW development, product development flow, lean thinking and field experience at enterprise scale
- Can be applied to organizations with a large number of practitioners and teams



<http://www.scaledagileframework.com/>

# SAFe for Lean Enterprises



## Module 4: Business & Technology Frameworks

95

© DevOps Institute unless otherwise stated



# Spotify Engineering Culture Part 1 with Henrik Kniberg (13:12)

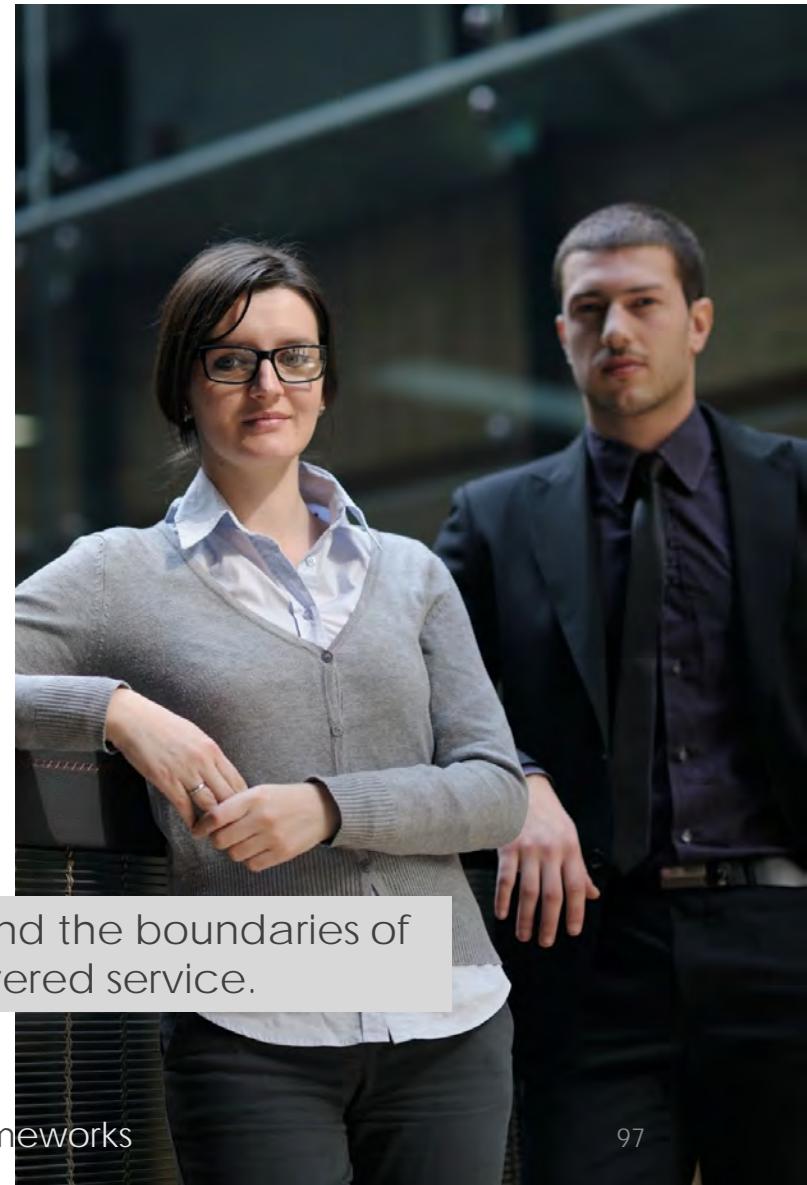
©DevOps Institute unless otherwise stated

## Module 4: Business & Technology Frameworks

# Increasing Agility

DevOps increases agility by:

- Breaking down silos
- Improving constraints
- Taking a unified approach to systems engineering
- Applying agile principles to both Dev and Ops
- Sharing knowledge, skills, experience and data
- Recognizing the criticality of automation
- Deploying faster with fewer errors



DevOps extends agile principles beyond the boundaries of the software to the entire delivered service.

Module 4: Business & Technology Frameworks

97

# IT Service Management (ITSM)



Gene Kim

# IT Service Management

IT service management (ITSM) is the implementation and management of quality IT services that meet the needs of the business.

- Provides guidance and structure to processes such as Change, Configuration, Release, Incident and Problem Management
- ITSM processes underpin the entire service lifecycle from strategy, design, transition, operations, continual improvement and value creation
- DevOps needs ITSM practices to meet the goal of deploying faster changes without causing disruption

Repeatable service management processes – adapted to an organization's current business needs – can lead the way to stable continuous delivery and increased flow.

# IT Infrastructure Library® (ITIL®)

ITIL® 4 defines a service as a means of enabling value co-creation by facilitating outcomes that customers want to achieve, without the customer having to manage specific costs and risks.

ITIL 4 consists of two key components:	
Four Dimensions Model	Service Value System
1. Organizations and people 2. Information and technology 3. Partners and suppliers 4. Value streams and processes	1. Guiding principles 2. Governance 3. Service value chain 4. Continual improvement 5. Practices

ITIL 4 (new in 2019) provides an emphasis on the business and technology world, how it works today, and how it will work in the future with Agile, DevOps and digital transformation.

AXELOS®, ITIL® and IT Infrastructure Library® are registered trademarks of AXELOS Limited.

## Module 4: Business & Technology Frameworks

101

©DevOps Institute unless otherwise stated

# ITIL® 4 Practices: DevOps Touches Them All

General Management Practices	Service Management Practices	Technical Management Practices
<ul style="list-style-type: none"><li>• Strategy management</li><li>• Portfolio management</li><li>• Architecture management</li><li>• Service financial management</li><li>• Workforce and talent management</li><li>• Continual improvement</li><li>• Measurement and reporting</li><li>• Risk management</li><li>• Information security management</li><li>• Knowledge management</li><li>• Organizational change management</li><li>• Project management</li><li>• Relationship management</li><li>• Supplier management</li></ul>	<ul style="list-style-type: none"><li>• Business analysis</li><li>• Service catalogue management</li><li>• Service design</li><li>• Service level management</li><li>• Availability management</li><li>• Capacity and performance management</li><li>• Service continuity management</li><li>• Monitoring and event management</li><li>• Service desk</li><li>• Incident management</li><li>• Service request management</li><li>• Problem management</li><li>• Release management</li><li>• Change enablement</li><li>• Service validation and testing</li><li>• Service configuration management</li><li>• IT asset management</li></ul>	<ul style="list-style-type: none"><li>• Deployment management</li><li>• Infrastructure and platform management</li><li>• Software development and management</li></ul>

All of these practices are needed in some form in DevOps – with the possible exception of Project Management.

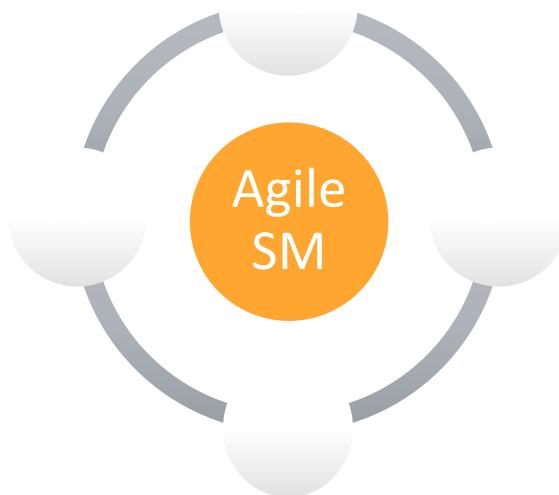
AXELOS®, ITIL® and IT Infrastructure Library® are registered trademarks of AXELOS Limited.

# DISCUSSION

## Agility in IT Operations

# Agile Service Management

Agile Service Management (Agile SM) ensures that ITSM processes reflect Agile values and are designed with “just enough” control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.



- Adapts Agile practices to ITSM process design
- Implements service management in small, integrated increments
- Ensures ITSM processes reflect Agile values from initial design through CSI
- Encourages “minimum viable” and “just enough” processes to increase speed and conformance

Source: Agile Service Management Guide

Agile Service Management does not reinvent ITSM – it modernizes the approach.

# Lean

Module 4: Business & Technology Frameworks

©DevOps Institute unless otherwise stated

# Lean Perspectives

DevOps has its roots in the lean manufacturing world, which addresses the problem of engineers designing products that factories can't afford to build.



# Sources of Waste = DOWNTIME

The goal of lean thinking is to create more value for customers with fewer resources and less waste.  
Waste is any activity that does not add value to the process.

Source	Purpose	Examples
Defects	Deviations from requirements; errors	Failures, known errors, misinformation
Overproduction	Producing more or faster than required	Excessive documentation or code
Waiting	Delays while waiting on a previous step	Delayed decisions, approvals, response
Non-use	Unused knowledge or creativity	Unused skill, innovation, communication
Transportation	Moving products from one location to another	Multiple hand-offs, emails or meetings
Inventory	Carrying more materials than needed	Unused software, infrastructure, excessive backlogs or emails
Motion	Moving people or assets more often than required	Moving code or infrastructure too much
Excessive processing	Doing more than is required	Over-engineering, failing to create templates and other reusable assets

# EXERCISE

## Identifying & Eradicating Waste

A black and white portrait of Gene Kim, a man with dark hair and glasses, smiling. He is wearing a striped shirt. To the right of the portrait is a dark gray speech bubble containing his quote.

“I believe that most of the DevOps patterns are the emergent properties that arise when you apply the techniques like Lean, the Toyota Production System, the Theory of Constraints and so forth to the IT value stream.”

Gene Kim

Module 4: Business & Technology Frameworks

©DevOps Institute unless otherwise stated.

# CASE STORY: Alaska Air

"We have to operate like a maverick brand; we have to operate fundamentally differently; we have to break the paradigm of what everyone thinks and imagines air travel is all about. So we focused on two key things: one – running an efficient operation and two – fostering technology innovation."



Veresh Sita, CIO

"Alaska views itself as a tech company with wings."

## Benefits

- Can securely expose its APIs to thousands of third party services
- Happier customers, higher revenues
- Scales easily and cost effectively
- Sites run at optimum point of performance and cost
- Better productivity, faster cross-pollination and knowledge sharing



# Value Stream Mapping

Value stream mapping is a lean tool that depicts the flow of information, materials and work across functional silos with an emphasis on quantifying waste, including time and quality.

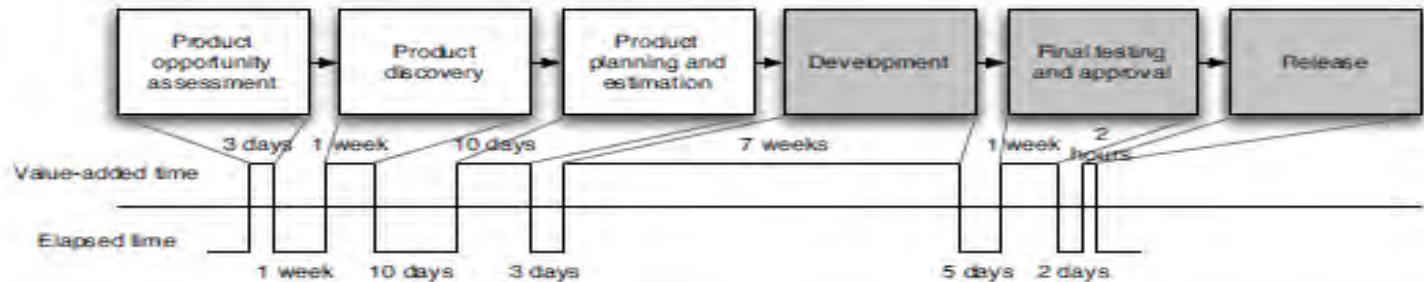
- A value stream is the sequence of activities required to design, produce, and deliver a specific product or service
- Value streams typically span multiple processes
- Value stream mapping enables cross-functional teams to:
  - See an entire value stream from a work and information flow perspective
  - Identify areas of non-value waste that could be eliminated in an effort to improve flow and deliver greater value
  - Identify, prioritize and measure improvements



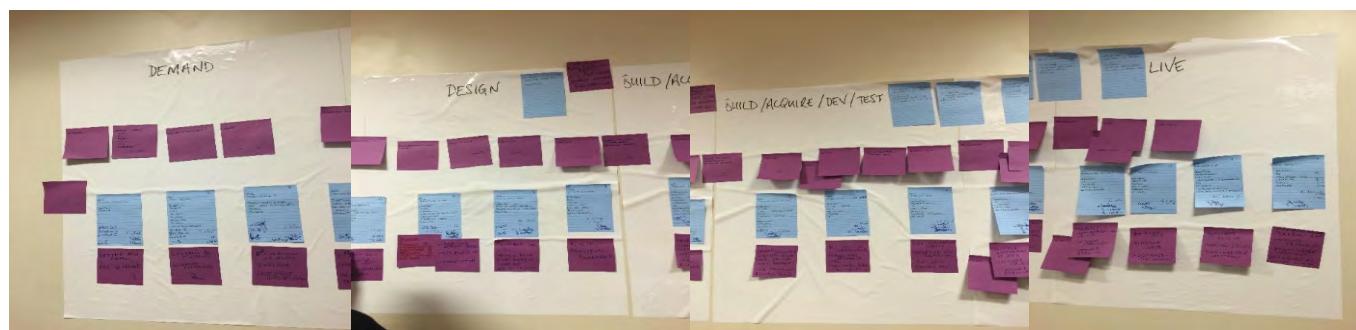
Module 4: Business & Technology Frameworks

111

# Sample Value Stream Maps



Source:  
Jez Humble - *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*

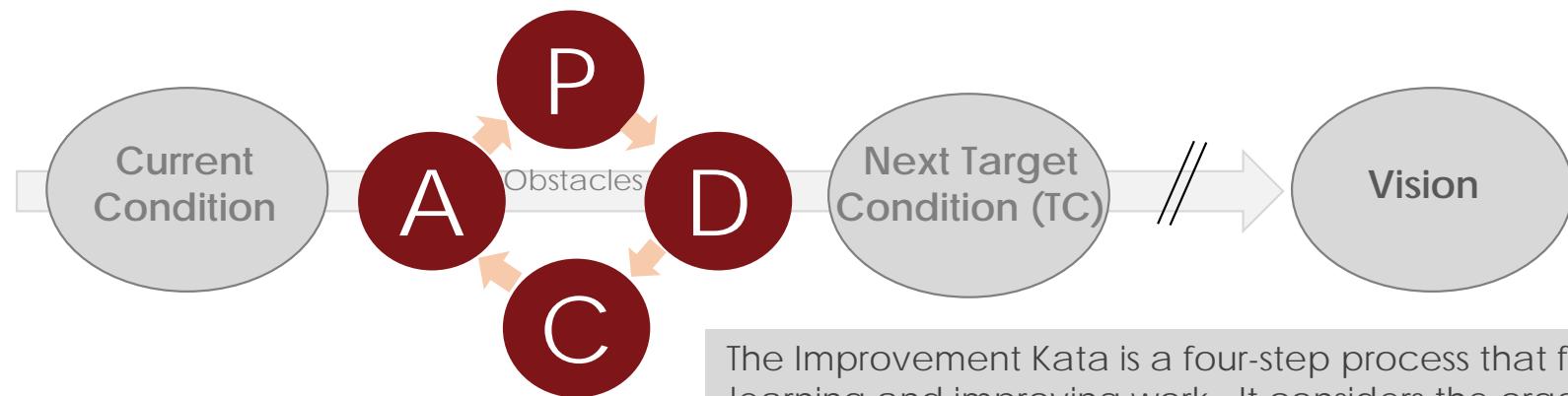


Source: Ranger4

# Improvement Kata

A kata is any structured way of thinking and acting that you practice until the pattern becomes a habit.

- 2 Grasp the current condition
- 4 PDCA and experiment toward the target condition
- 3 Establish the next target condition
- 1 Understand the long-term vision or direction



The Improvement Kata is a four-step process that focuses on learning and improving work. It considers the organization's long-term vision or direction.  
Plan > Do > Check > Act (PDCA)

Module 4: Business & Technology Frameworks

©DevOps Institute unless otherwise stated

# Safety Culture

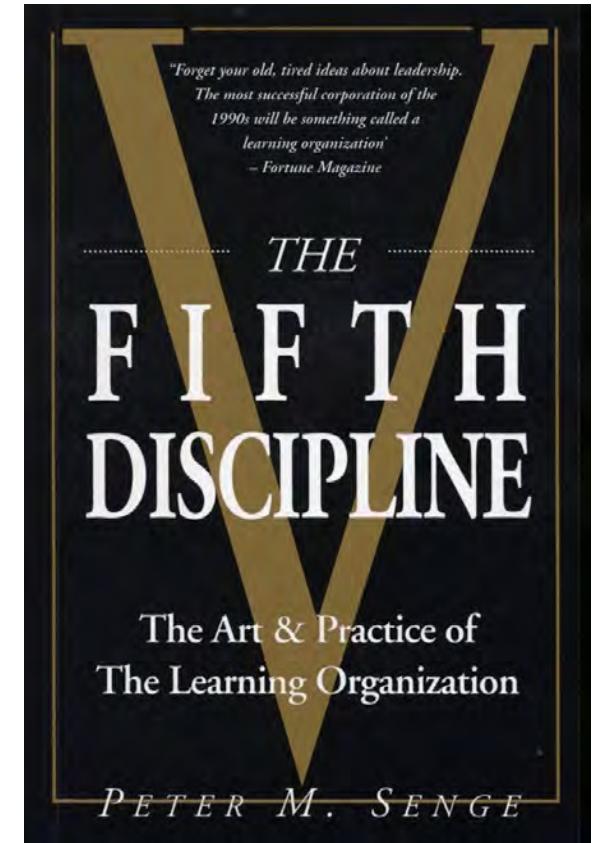
- Attitude, beliefs, perceptions and values that employees share in relation to safety in the workplace
- Blameless postmortems
- Valuing incidents
- Avoiding Single Points of Failures (SPOFs)
- The Andon Cord – thank you for creating a learning opportunity

“An incident is an unplanned investment, and if you don't see it that way as a leader, you are not getting a return on the investment that was already made on your behalf.”

Attributed to John Allspaw by Sidney Dekker in Beyond the Phoenix Project

# Learning Organizations

- Have a commitment to learning
- Improvement requires learning something new
- Not learning creates cultural debt
- Humans love mastery (and autonomy and purpose)
- Management commitment is essential



Module 4: Business & Technology Frameworks

115

# Continuous Funding

- Traditional funding happens on annual cycles
- Agile funding can be:
  - Fixed cost or continuous
  - Frequently reviewed
- Product/Team based funding
- Venture (or bet based) funding
- Focus on measuring return

In reply to Helen Beal



**Jonathan Smart**  
@jonsmart

@HelenRanger4 we're starting to pilot agile investment with qtrly rolling wave instead of annual budgeting.

30/06/2016, 11:44 from Paddington, London



# Module 4: Quiz

- 1 In the Agile Manifesto, we value working software over:
  - a) Processes and tools
  - b) Comprehensive documentation
  - c) Contract negotiations
  - d) Following a plan
- 2 Which of these is not an ITSM process model?
  - a) Change model
  - b) Release model
  - c) Incident model
  - d) Development model
- 3 Which of these is not a Lean tool?
  - a) A5 thinking
  - b) Value Stream Mapping
  - c) Improvement kata
  - d) Kanban
- 4 What is the first step in the improvement kata?
  - a) Grasp the current condition
  - b) Establish the next target condition
  - c) Plan Do Check Act (PDCA)
  - d) Understand the long term vision or direction
- 5 Who 'wrote the book' on Learning Organizations?
  - a) Peter Senge
  - b) Jonathan Smart
  - c) Henrik Kniberg
  - d) Gene Kim

# Module 4: Quiz Answers

- |   |   |   |
|---|---|---|
| 1 | In the Agile Manifesto, we value working software over: | a) Processes and tools<br><b>b) Comprehensive documentation</b><br>c) Contract negotiations<br>d) Following a plan  |
| 2 | Which of these is not an ITSM process model?            | a) Change model<br>b) Release model<br>c) Incident model<br><b>d) Development model</b>   |
| 3 | Which of these is not a Lean tool?                      | <b>a) A5 thinking</b><br>b) Value Stream Mapping<br>c) Improvement kata<br>d) Kanban  |
| 4 | What is the first step in the improvement kata?         | a) Grasp the current condition<br>b) Establish the next target condition<br>c) Plan Do Check Act (PDCA)<br><b>d) Understand the long term vision or direction</b> |
| 5 | Who 'wrote the book' on Learning Organizations?         | <b>a) Peter Senge</b><br>b) Jonathan Smart<br>c) Henrik Kniberg<br>d) Gene Kim  |

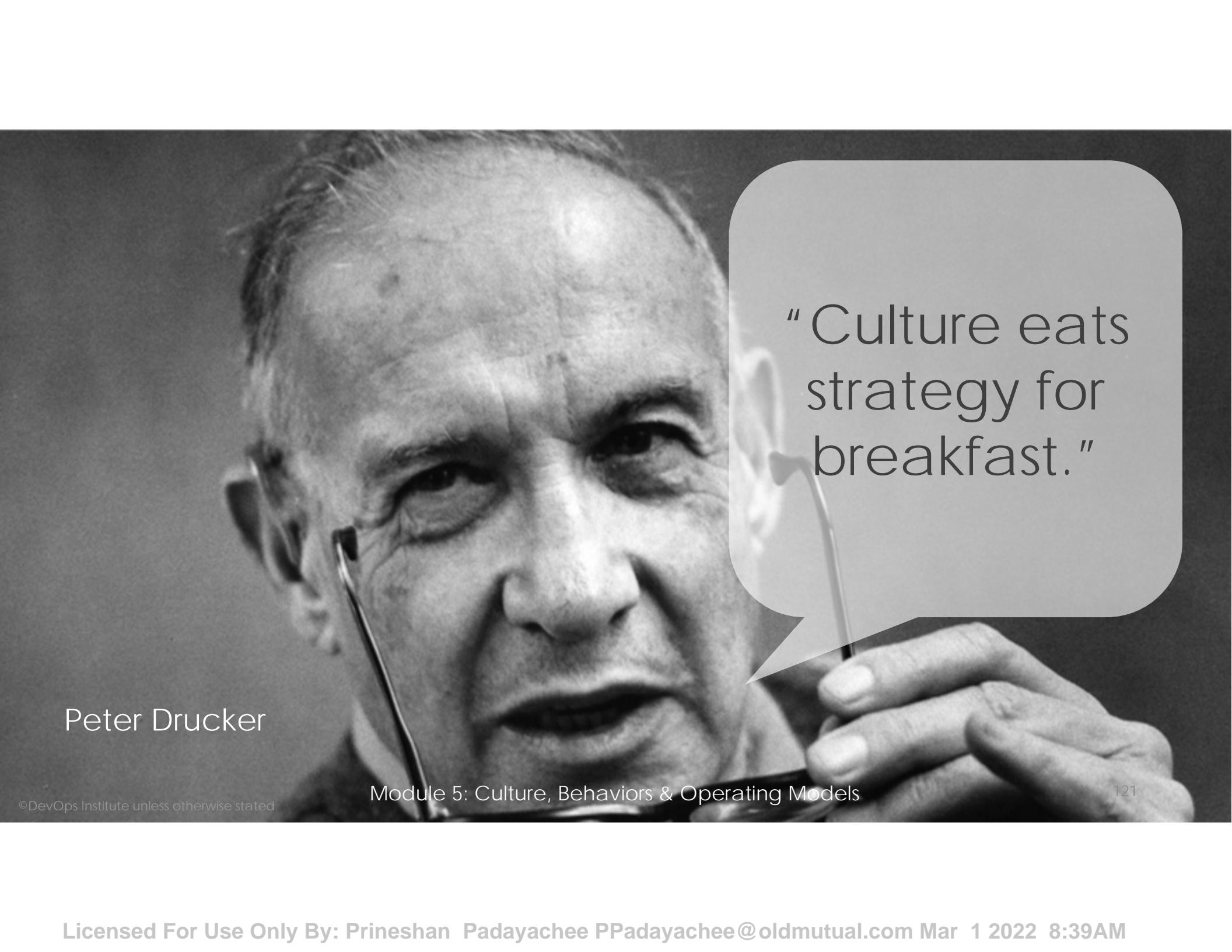
# Module 5

## CULTURE, BEHAVIORS & OPERATING MODELS

# Module 5: Culture, Behaviors & Operating Models

- Defining Culture
- Cultural Debt
- Behavioral Models
- Organizational Models

Component	Module 5 Content
Video	Spotify Engineering Culture Part 2
Case Story	Target
Discussion	Placing on the Change Curve
Exercise	Rating & Improving Using the Westrum Model

A black and white close-up photograph of Peter Drucker's face. He is looking slightly downwards and to his right with a thoughtful expression. He has thinning hair and is wearing glasses. A speech bubble originates from his mouth, containing a quote.

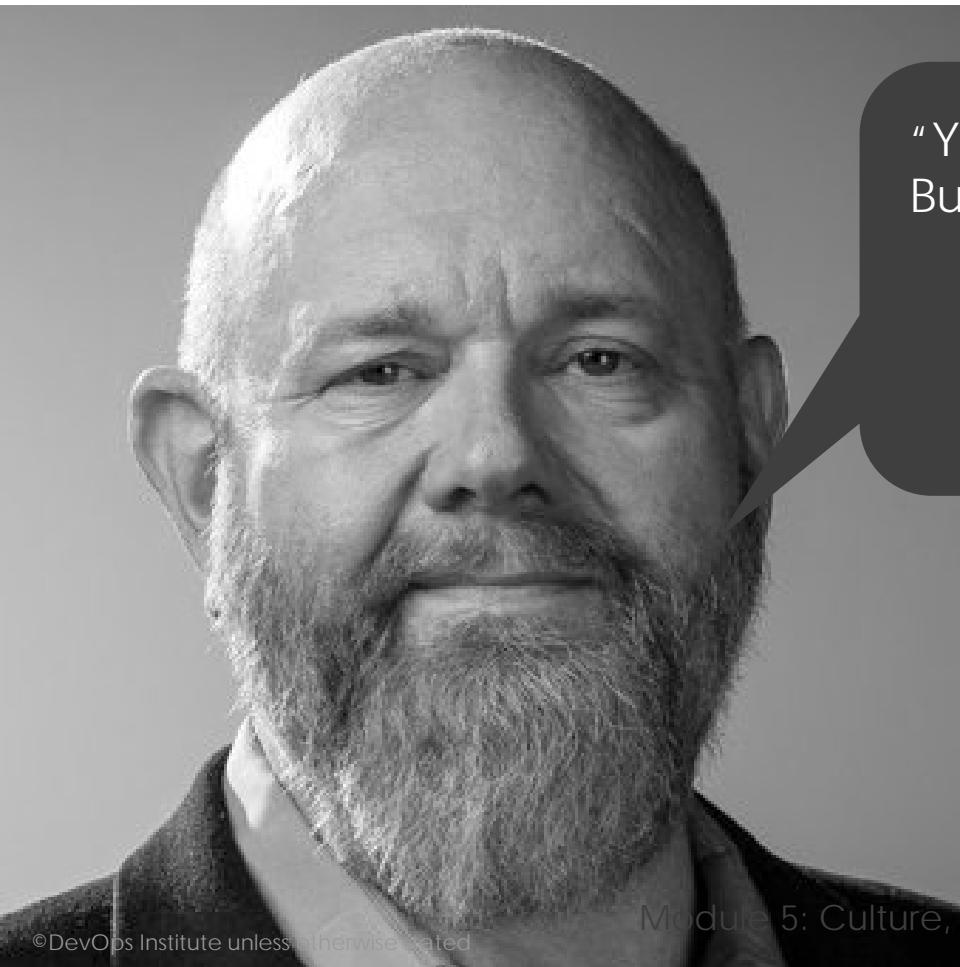
“Culture eats  
strategy for  
breakfast.”

Peter Drucker

Module 5: Culture, Behaviors & Operating Models

121

# What is Organizational Culture?



“You can’t directly change culture. But you can change behavior, and behavior becomes culture.”

Lloyd Taylor, VP IT Operations  
LinkedIn

The values and behaviors that contribute to the unique social and psychological environment of an organization.

[www.businessdictionary.com](http://www.businessdictionary.com)

# DevOps Helps to Overcome Cultural Debt

Cultural debt occurs when cultural considerations are disregarded or deferred in favor of growth and innovation.



“The effective interest rate on cultural debt is usually higher than on technical debt.”

Dharmesh Shah, Founder & CTO



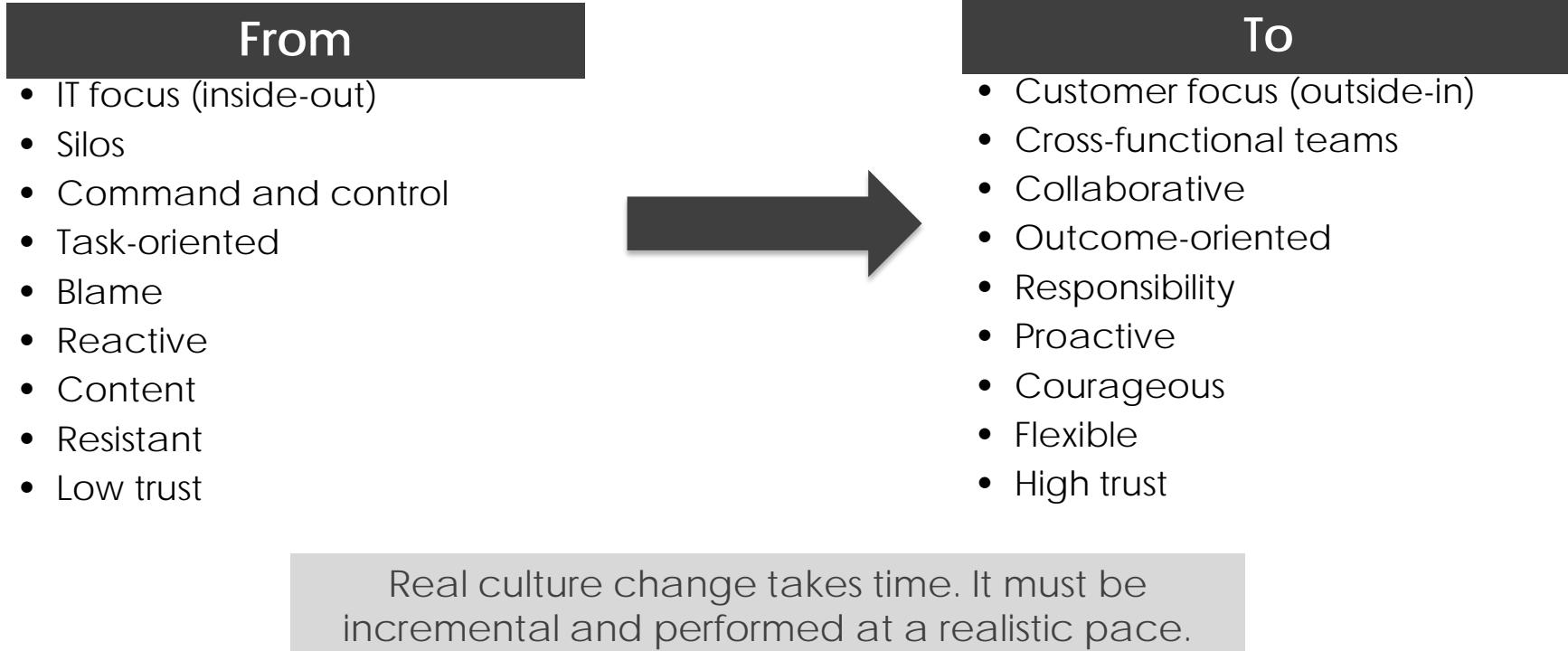
IT's silo culture and other organizational challenges are a direct result of disregarding cultural considerations in favor of rapid increases in corporate technology. The due date is today!

# Characteristics of a DevOps Culture

- Shared vision, goals and incentives
- Open, honest, two-way communication
- Collaboration
- Pride of workmanship
- Respect
- Trust
- Transparency
- Continuous improvement
  - Experimentation
  - Intelligent risk taking
  - Learning and practicing
- Data-driven
- Safe
- Reflection
- Recognition

Organizational culture is one of the strongest predictors of both IT performance and overall performance of the organization.

# Shifting Thoughts and Behaviors

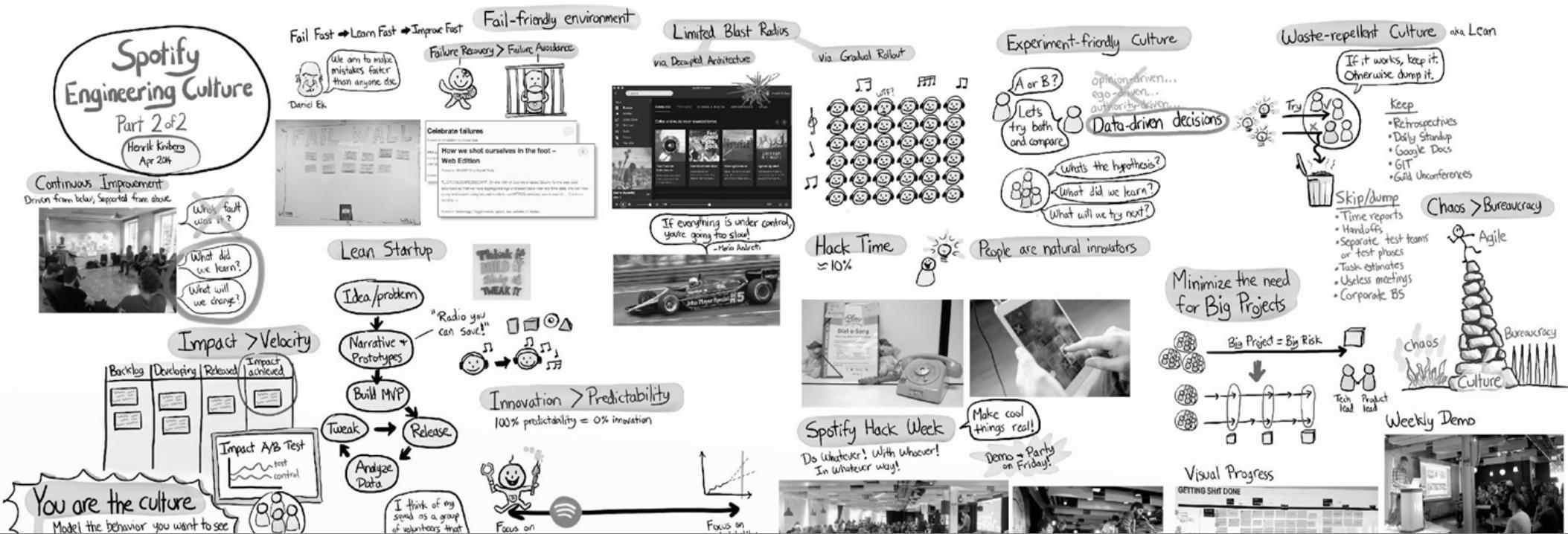


# High Trust vs. Low Trust



Module 5: Culture, Behaviors & Operating Models

126



# Spotify Engineering Culture Part 2 with Henrik Kniberg (13:27)

# Culture and the Flow of Information

Pathological (Power-oriented)	Bureaucratic (Rule-oriented)	Generative (Performance-oriented)
Information is hidden	Information may be ignored	Information is actively sought
Messengers are 'shot'	Messengers are isolated	Messengers are trained
Responsibilities are shirked	Responsibility is compartmentalized	Responsibilities are shared
Bridging is discouraged	Bridging is allowed but discouraged	Bridging is rewarded
Failure is covered up	Organization is just and merciful	Failure causes enquiry
Novelty is crushed	Novelty creates problems	Novelty is implemented

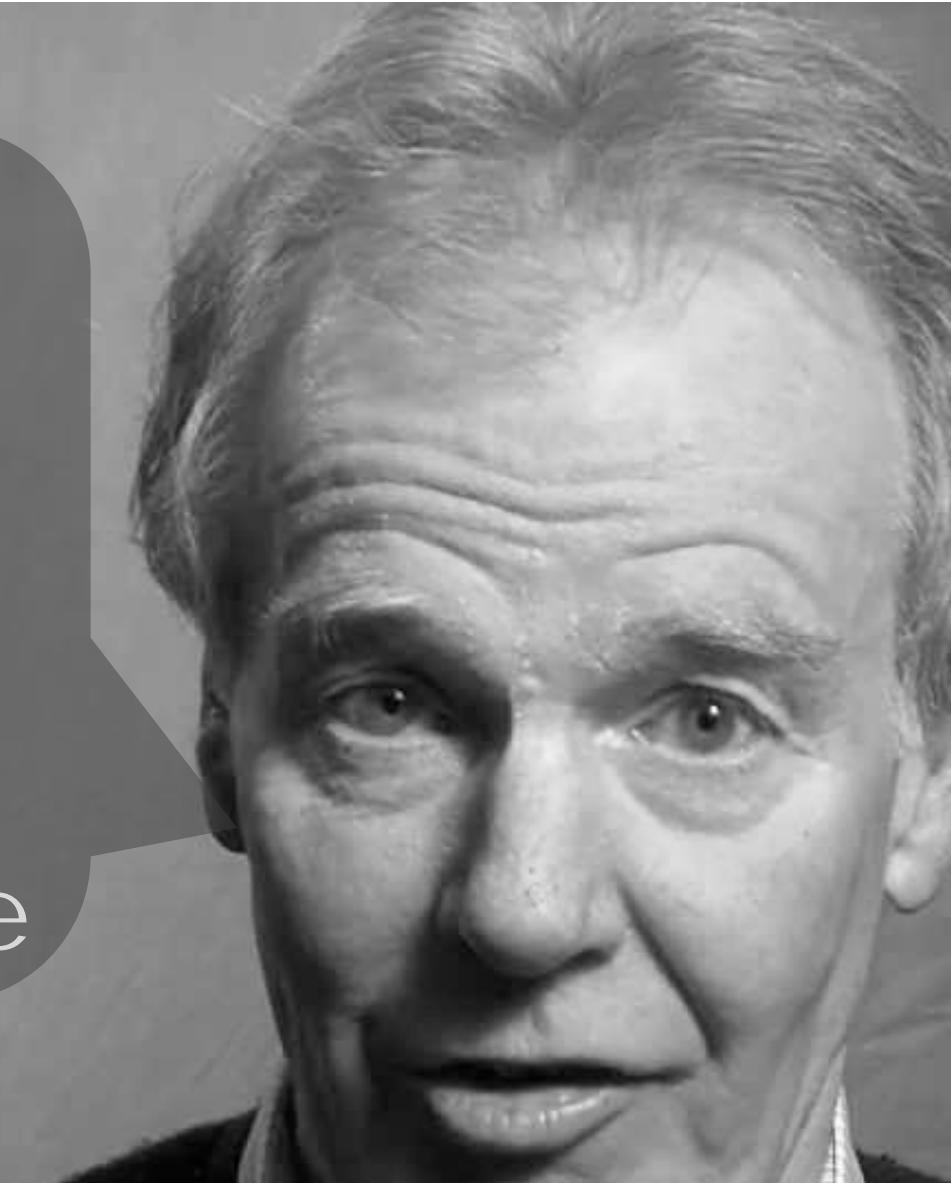
Source: Westrum, *A Typology of Organizational Cultures*

High-trust organizations encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas.

## Module 5: CULTURE, BEHAVIORS & OPERATING MODELS

### EXERCISE

Rating & Improving Using the Westrum Model



“People don't  
resist change.  
They resist being  
changed.”

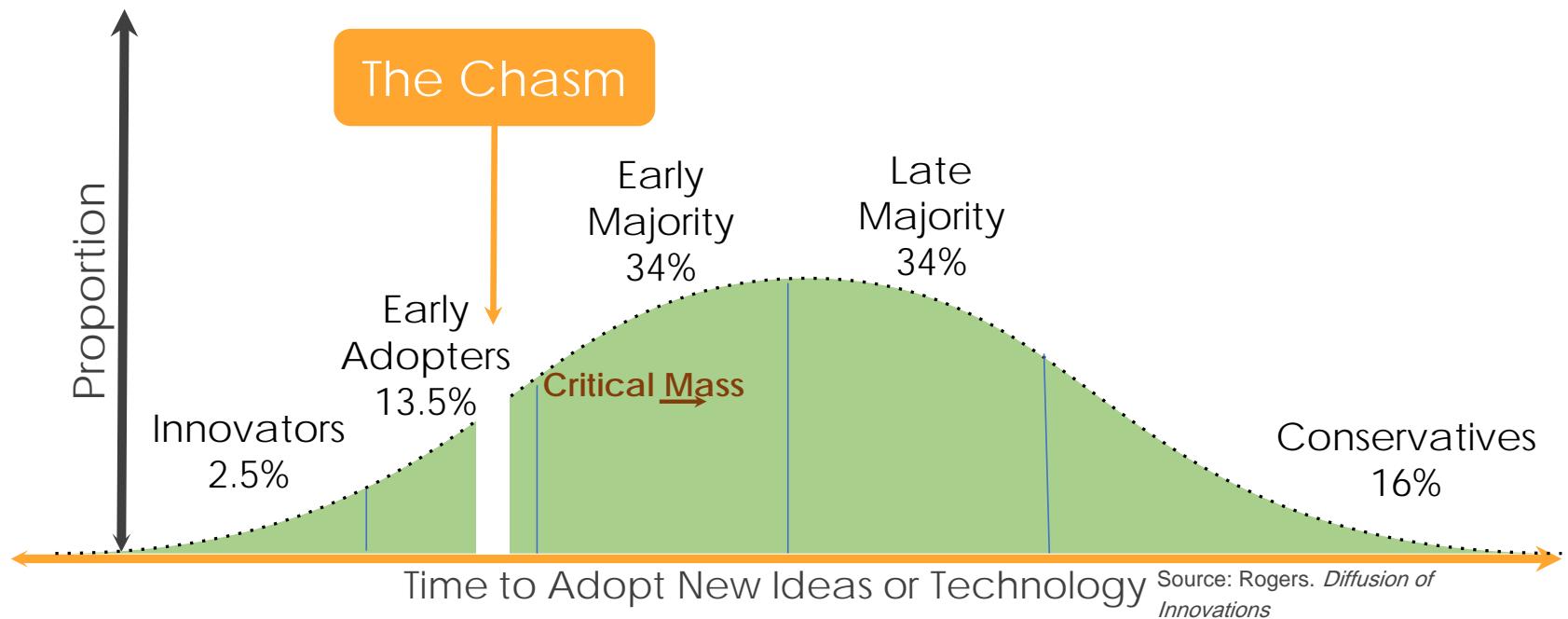
Peter Senge

# Culture Change is Never Easy

People typically don't resist their own ideas.

- You can't change people; they can only change themselves
- Change almost always takes longer and costs more than expected
- Stakeholder involvement is critical
- People who participate in what and how to change decisions are far more likely to accept change

# People Adapt to Change at Different Paces



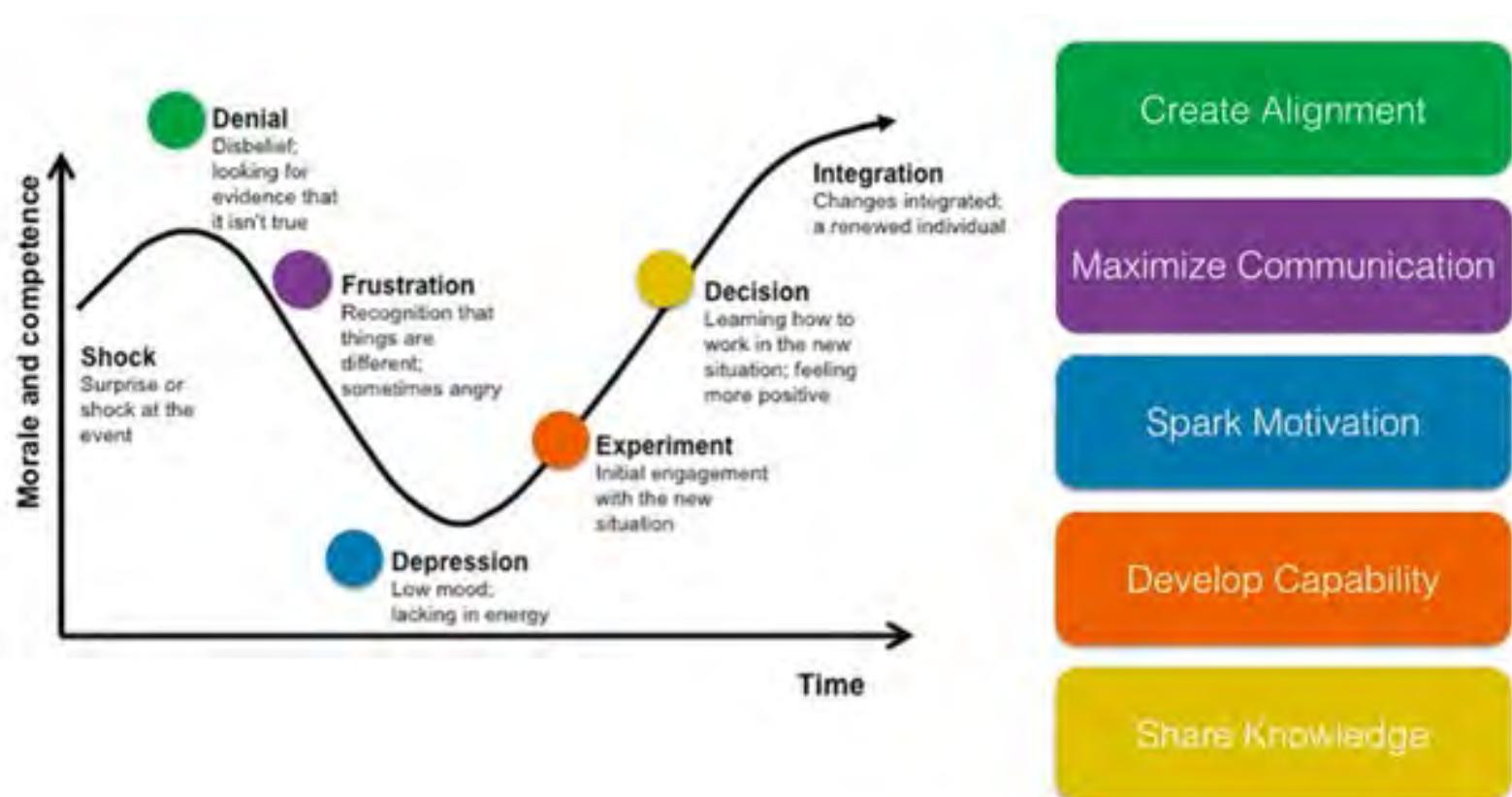
Adoption means that a person does something differently than before.

## Module 5: CULTURE, BEHAVIORS & OPERATING MODELS

### DISCUSSION

#### Placing on the Change Curve

# The Stages of Change Acceptance



The Kübler-Ross Change Curve

From [leanchange.org](http://leanchange.org)

# Communication is Critical

- A DevOps culture requires timely and effective communication
- Shared tools facilitate timely and meaningful communication
  - Chat platforms
  - Task managers
  - Social tools
  - Alert management tools
  - Knowledge sharing platforms

# Encourage Collaborative Relationships

Collaboration involves people jointly working with others towards a common goal. In a collaborative environment, each person's contribution is valued.

- Collaboration
  - Is voluntary (ideally)
  - Involves sharing
    - Responsibility for outcomes
    - Resources
  - Requires cooperation, respect and trust
- Requires participation
  - Providing feedback
  - Identifying and solving problems
  - Learning and sharing knowledge and expertise
  - Sharing and even swapping responsibilities
  - Making and keeping realistic commitments

What's the difference between collaboration and communication?

# Expect Some Conflict: Thomas-Kilmann Conflict Modes

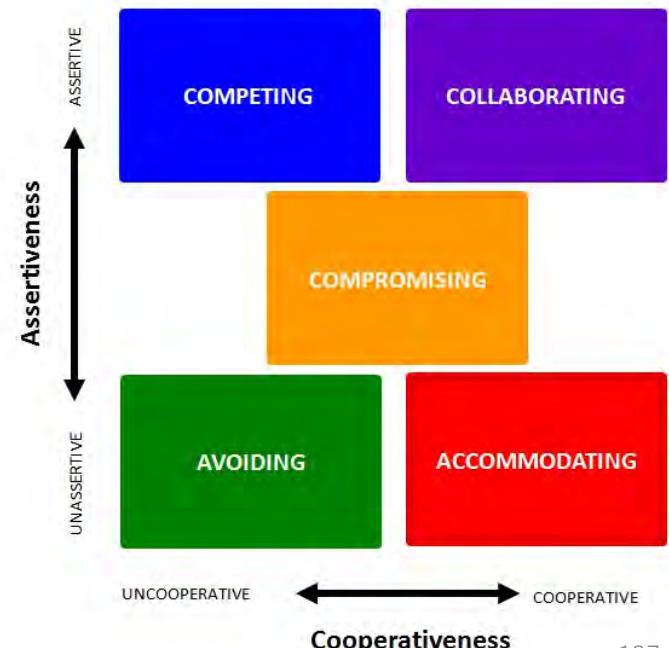
Because no two individuals have exactly the same expectations and desires, conflict is a natural part of our interactions with others.

The Thomas-Kilmann Conflict Inventory (TKI) measures a person's behavioral choices under certain conflict situations.

Conflict Mode	Approach	Result
Competing	Assertive and Uncooperative	Win/Lose
Collaborating	Assertive and Cooperative	Win/Win
Compromising	Partially Assertive and Cooperative	Each Wins and Loses
Avoiding	Unassertive and Uncooperative	Lose/Lose
Accommodating	Unassertive and Cooperative	Lose/Win

Source: [www.diagnostics.com](http://www.diagnostics.com)

**Thomas-Kilmann Conflict Mode Instrument**



# Avoid Change Fatigue

Change fatigue is a general sense of apathy or passive resignation towards organizational changes by individuals or teams.

- View resistance to change as normal
  - Listen, empathize
- Communicate the big picture
  - Explain the reason for *this* change
  - Show how changes are connected
  - Tie changes to business strategies and goals
- Ensure each change initiative has an intended outcome
- Empower people to contribute
- Celebrate (even if only small) successes
- Create visible feedback and improvement loops

The amount of change fatigue that people experience is directly impacted by the way change is managed.

# Empower New Behaviors

- Improve communication and collaboration practices and shared tools
- Create a common vocabulary
- Job shadowing
- Cross-skilling
- Immersion experiences
- Team building
- Communities of practice
- Internal DevOps Days
- Game days (hackathons)
- Simulations
- Social-media style idea and story sharing and problem solving

Sharing between peers, organizations and industries is a crucial factor in the growth and acceptance of DevOps.

# CASE STORY: Target

"When we asked for permission we were told no, but we did it anyways because we knew we needed to. We ran tools hackathons alongside our internal DevOpsDays events and we hosted a ton of meetups. We've hosted 6 internal DevOpsDays events."

"We are a technology company."



Heather Mickman,  
Transformative  
Technology  
Executive

Ross Clanton, Head  
of Engineering

## Benefits

- Made structural changes gaining bottom-up, then top-down support
- Converged the agile and DevOps efforts
- Used training, coaching and immersive experiences – massive Dojo!
- Built a full stack environment in minutes instead of 3-6 months
- Built empathy and understanding

# Module 5: Quiz

- |   |   |  |
|---|---|--|
| 1 | Who said: "Culture eats strategy for breakfast?"  | a) Peter Drucker<br>b) Gene Kim<br>c) Damon Edwards<br>d) Bill Gates   |
| 2 | What can't you change?                            | a) Behavior<br>b) Habits<br>c) Culture<br>d) Systems   |
| 3 | What is a characteristic of a DevOps culture?     | a) Blame<br>b) Mistrust<br>c) Fear<br>d) Courage   |
| 4 | Which of these happens in a pathological culture? | a) Messengers are shot<br>b) Responsibilities are shared<br>c) Failure causes enquiry<br>d) Novelty is implemented |
| 5 | What didn't Target do?                            | a) Run hackathons<br>b) Set up Dojos<br>c) Get permission<br>d) Build empathy                                      |

# Module 5: Quiz Answers

- |   |   |  |
|---|---|--|
| 1 | Who said: "Culture eats strategy for breakfast"?  | a) Peter Drucker<br>b) Gene Kim<br>c) Damon Edwards<br>d) Bill Gates   |
| 2 | What can't you change?                            | a) Behavior<br>b) Habits<br>c) Culture<br>d) Systems   |
| 3 | What is a characteristic of a DevOps culture?     | a) Blame<br>b) Mistrust<br>c) Fear<br>d) Courage   |
| 4 | Which of these happens in a pathological culture? | a) Messengers are shot<br>b) Responsibilities are shared<br>c) Failure causes enquiry<br>d) Novelty is implemented |
| 5 | What didn't Target do?                            | a) Run hackathons<br>b) Set up Dojos<br>c) Get permission<br>d) Build empathy                                      |

# Module 6

## AUTOMATION & ARCHITECTING DEVOPS TOOLCHAINS

# Module 6: Automation & Architecting DevOps Toolchains

- CI/CD
- Infrastructure as Code
- Cloud
- Containers & Microservices
- Machine Learning
- DevOps Toolchains

Component	Module 6 Content
Video	The DevOps Toolchain with John Okoro
Case Story	Fannie Mae
Discussion	Applying the DevOps Handbook's Definition
Exercise	Architect Your DevOps Toolchain



"DevOps is not about automation, just as astronomy is not about telescopes."

Christopher Little, quoted in The DevOps Handbook

# The Periodic Table of DevOps Tools (V4.2)

 digital.ai™

CollabNetVersionOne, XebiaLabs, Arxan, Numerify & Experitest  
are now Digital.ai

91	Os	92	En	93	Os	94	Os	95	Fm	96	Os	97	Pd	98	Os	99	En	100	Os	101	En	102	Pd	103	En	104	Pd	105	Os		
Jn	Azure DevOps Code	Azc	Glc	Tr	CircleCI	Cc	Mv	Ab	Gd	Acb	Aj	Bi	At	Sw	Td	Pd	Jenkins	GitLab CI	Travis CI	Atlassian Bamboo	Gradle	AWS CodeBuild	Atlassian Jira	BMC Helix ITSM	Atlassian Trifecta	ServiceNow	TQdesk	PagerDuty			
106	Fr	107	Pd	108	Fr	109	Fr	110	Pd	111	En	112	En	113	Os	114	Fr	115	Fr	116	Pd	117	En	118	En	119	En	120	Os		
Tt	Nn	Se	Ju	Sl	Ct	Ap	Sq	Cu	Jm	Pa	Dai	Tp	Pr	Gl	Tricentis Tosca	Neotys Neoload	Selenium	JUnit	Sauce Labs	Compuware Topaz	Appium	Squash TM	Cucumber	JMeter	Parasoft	Digital.ai	Tasktop	Plutora	GitLab		

# Automation Benefits

Automation supports:

- Faster lead times
- More frequent releases
- Less turbulent releases
- Fewer errors
- Higher quality
- Improved security and risk mitigation
- Faster recovery
- Business and customer satisfaction

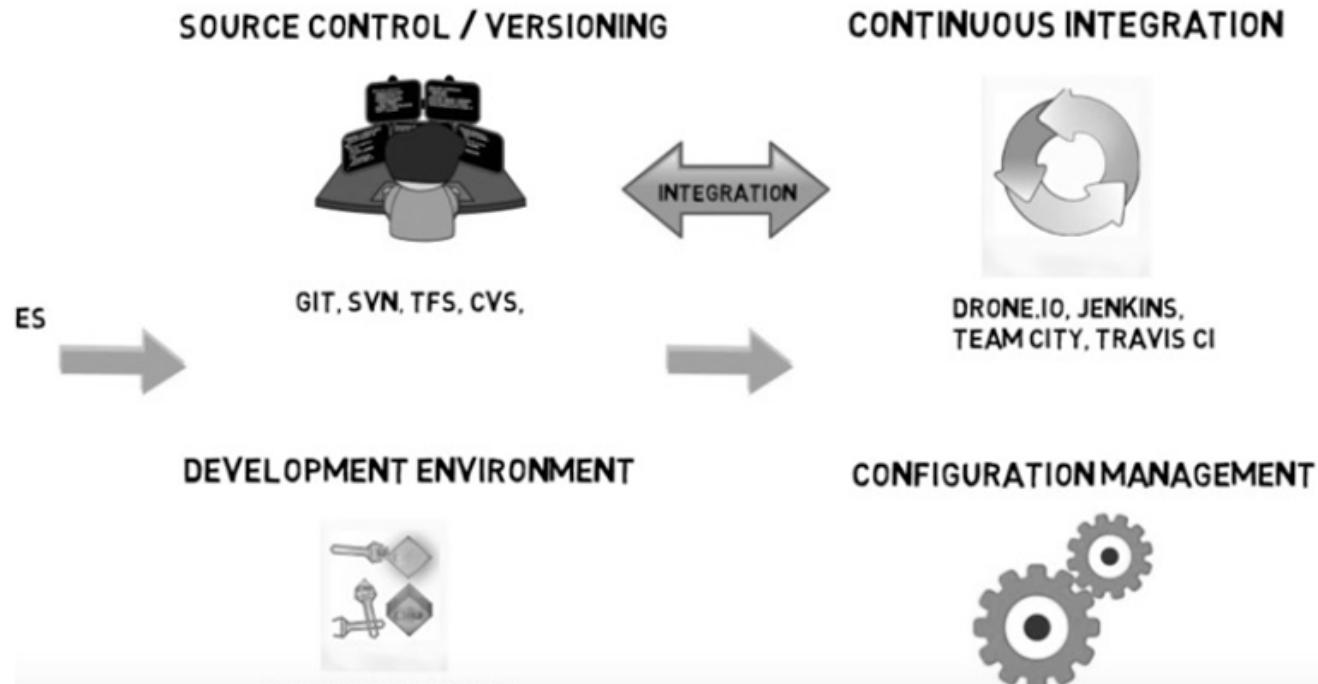
Automation gives rote tasks to computers and allows people to:

- Weigh evidence
- Solve problems
- Make decisions based on feedback
- Use their skills, experience and judgment

*“Your tools alone will not make you successful.”*

Patrick Debois

Up next



# The DevOps Toolchain with John Okoro (7:43)



© DevOps Institute Published on May 28, 2016

Module 6: Automation & Architecting DevOps Toolchains

SUBSCRIBED 715



# Important Terms

- **Artifact**

- Any element in a software development project including documentation, test plans, images, data files and executable modules

- **Application Programming Interface (API)**

- A set of protocols used to create applications for a specific OS or as an interface between modules or applications

- **Microservices:**

- A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system. This is known as loose coupling

- **Operating System (OS) Virtualization**

- A method for splitting a server into multiple partitions called "containers" or "virtual environments" in order to prevent applications from interfering with each other

- **Containers**

- A way of packaging software into lightweight, stand-alone, executable packages including everything needed to run it (code, runtime, system tools, system libraries, settings) for development, shipment and deployment.

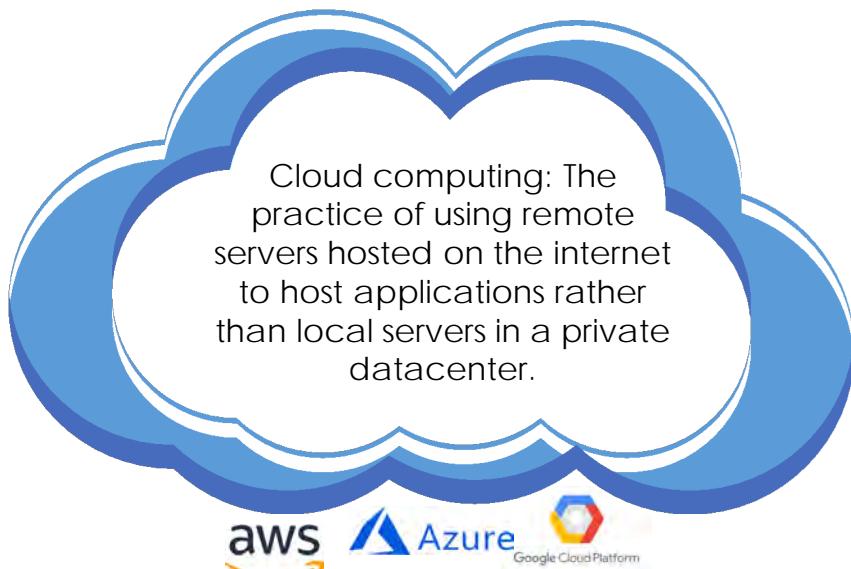
- **Open source**

- Software that is distributed with its source code so that end user organizations and vendors can modify it for their own purposes

- **Machine Learning**

- Data analysis that uses algorithms that learn from data

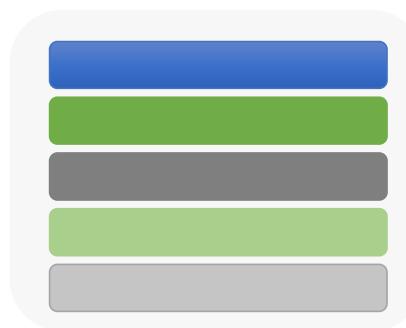
# Cloud, Containers and Microservices



**docker** is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package



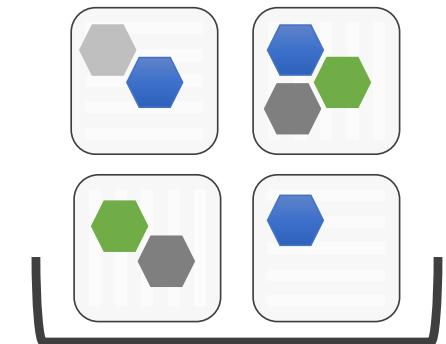
Elite performers were **24 times** more likely to have met all essential cloud characteristics than low performers.



Monolithic



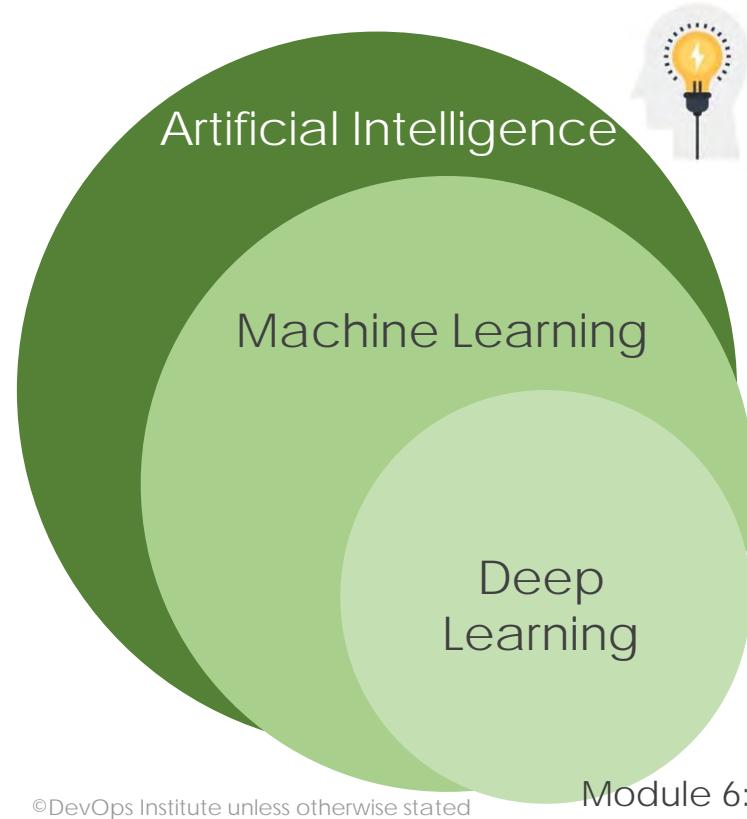
**kubernetes** is an open source system for managing containerized applications across multiple hosts, providing basic mechanisms for deployment, maintenance, and scaling of applications



Containers & Microservices

# AI & Machine Learning

Giving computers the ability to "learn" with data, without being explicitly programmed.



- Organizations are collecting more data than ever
- It's hard to fully extract the value from that data
- Data science is an increasingly popular discipline
- AI and Machine Learning enables predictive analytics
- Can find trends and correlations humans can't
- Augments human contribution
- Boosts productivity
- Automated feedback loops

Definition: Data analysis that uses algorithms to learn from data.

# DevOps Automation Practices

A tool chain philosophy involves using an integrated set of complimentary task specific tools to automate end-to-end delivery and deployment processes.

- Tool chain (vs. a single-vendor solution)
- Shared tools
- Self-service
- Architecting software in a way that enables
  - Test automation
  - Monitoring
- Infrastructure as Code
- Experimentation

Avoid tools that enforce silos!

# Communication and Collaboration Can Be Automated Too

Innovative tools and platforms facilitate and expedite communication and collaboration across the Dev and Ops spectrum.

How to	Tools
<ul style="list-style-type: none"><li>• Issue alerts and alarms</li><li>• Improve response</li><li>• Provide at a glance status updates</li><li>• Improve workflow</li><li>• Improve information flow</li><li>• Enable virtual collaboration</li><li>• Enable cross-functional, cross-skilling and job sharing</li></ul>	<ul style="list-style-type: none"><li>• Communication platforms</li><li>• Dashboards</li><li>• Kanban boards</li><li>• Group chat rooms (ChatOps)</li><li>• Workflow and project management tools</li><li>• Document sharing</li><li>• Wikis and knowledge management systems</li><li>• ITSM tools</li><li>• Social tools</li><li>• Shared backlogs</li></ul>

# First Steps to Improving DevOps Automation

- Architect before automating
- Assess your existing tools and automation capabilities
- Simplify first – don't automate bad processes
- Identify critical gaps
- Seek vendors who can meet *your* requirements
- Automate high value, repetitive and error-prone work
- Optimize workflow bottlenecks and communication
- Improve automated monitoring and notification practices
- Expect this to be an iterative process - your toolchain will evolve over time

Do not underestimate the effort and cost of building toolchains from open source applications. Open source is not necessarily free. It means that you can modify the source to fit your needs.

# CASE STORY: Fannie Mae

"We drive adoption as it makes sense on an app-by-app basis. It's been going on for about a year and a half and we're reaching a critical mass point where people are really lining up. We can be much more flexible, much more dynamic, and provide our customers and partners with the tools that they need to interact with us far more easily. Like anybody else, we've got to get ideas to production a lot faster than we're getting them there today."

"We're putting ourselves in a position to be much easier to work with."



Jason Anders,  
IT Leadership for  
Securitisation

## Benefits

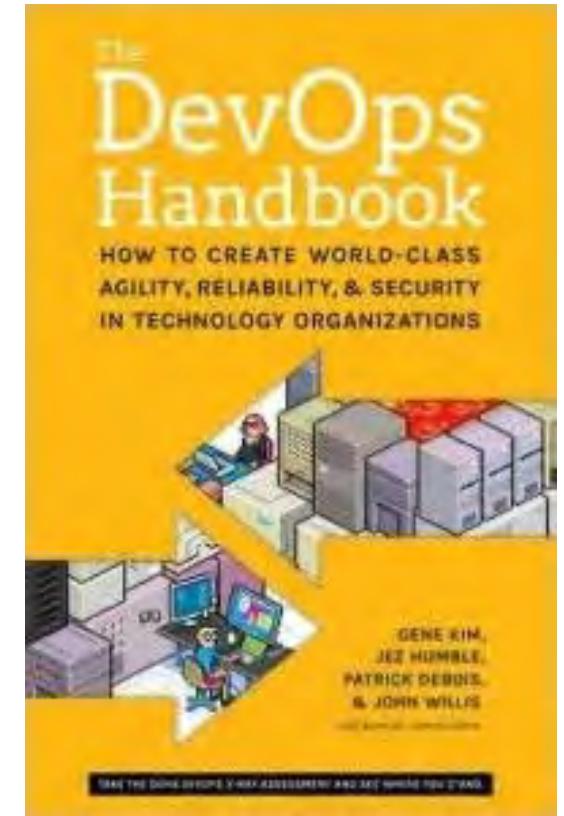
3-day deployment reduced to 45 minutes

- Deploys seven or eight times a day
- 40-75% savings on storage costs thanks to data virtualization

# DevOps Toolchains

“One way to enable market-oriented outcomes is for Operations to create a set of centralized platforms and tooling services that any Dev team can use to become more productive... a platform that provides a shared version control repository with pre-blessed security libraries, a deployment pipeline that automatically runs code quality and security scanning tools, which deploys our applications into known, good environments that already have production monitoring tools installed on them.”

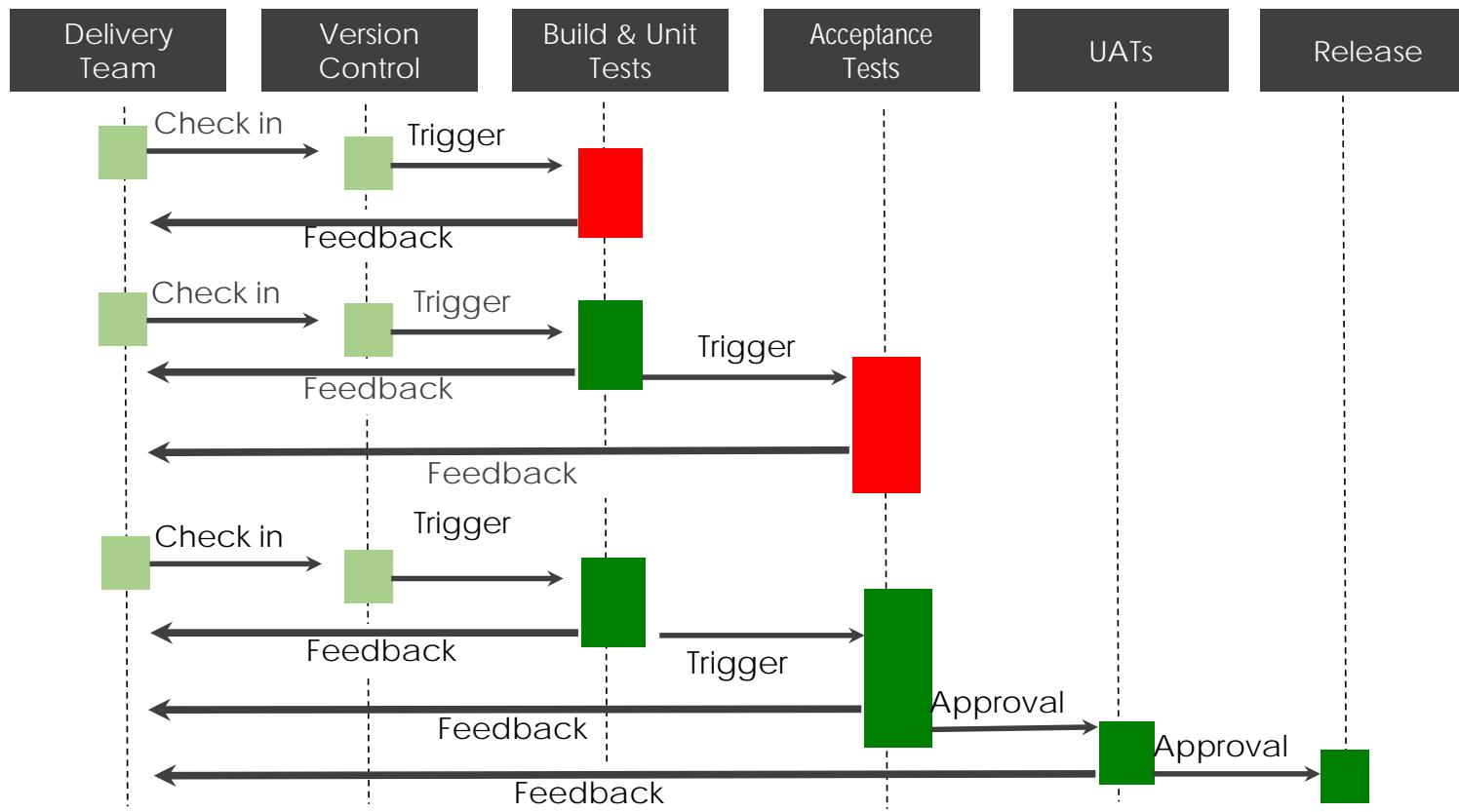
## The DevOps Handbook



## DISCUSSION

# Applying the DevOps Handbook's Definition

# The Deployment Pipeline



The deployment pipeline is an automated process for managing all changes, from check-in to release. Toolchains span silos and automate the deployment pipeline.

Source: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation

159

# DevOps Toolchains

The DevOps toolchain is composed of the tools needed to support a DevOps continuous integration, continuous deployment, and continuous release and operations initiative. (Gartner)

- Toolchains automate tasks in the deployment pipeline
- Each element of the toolchain serves a specific purpose
- Applications within the toolchains are connected via APIs
- They do not have to be homogenous or from a single vendor
- Toolchains are usually built around open and closed source ecosystems
- Require an architectural design to ensure interoperability and consistency

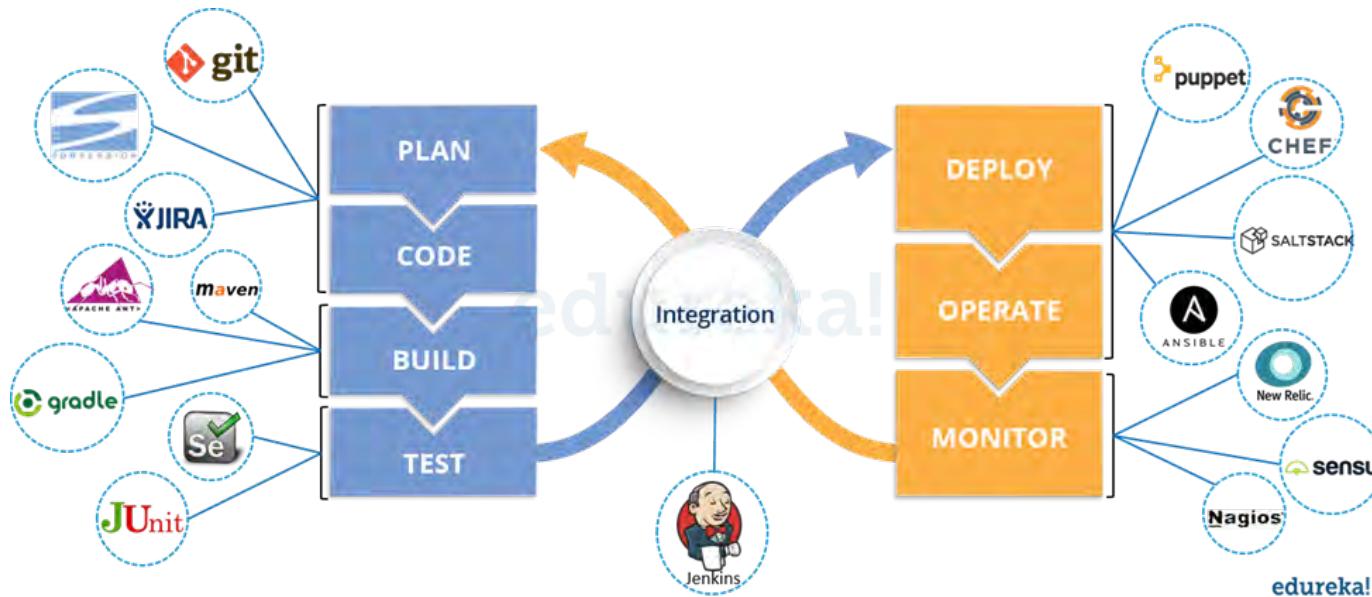
The deployment pipeline is an automated process for managing all changes, from check-in to release. Toolchains span silos and automate the deployment pipeline.

How should DevOps toolchains interface to operational tools such as monitoring or support applications?

Source: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*

# Sample DevOps Toolchain (US Government - GSA)

There are many established open- and closed-source DevOps-enabled tools with vibrant ecosystems.



How these tools are adapted and integrated into your deployment pipeline will determine their value.

# Elements in a DevOps Toolchain

- The deployment pipeline breaks the software delivery lifecycle into logical stages
- Each stage provides
  - The opportunity to verify the quality of new features from a different angle
  - The team with fast feedback
  - Visibility into the flow of changes
- DevOps toolchains provide the capabilities needed to automate and expedite each stage

## Typical Toolchain Elements:

- Requirements management
- Orchestration and visualization
- Version control management
- Continuous integration and builds
- Artifact management
- Containers and OS virtualization
- Test and environment automation
- Server configuration and deployment
- System configuration management
- Alerts and alarms
- Monitoring

## EXERCISE

# Architect Your DevOps Toolchain

# Build Your DevOps Toolchain Gradually

Do not create a definitive toolchain that applies to all DevOps projects. The toolchain is a foundation requiring continuous innovation and customization to meet your specific and ongoing DevOps priorities.

Model your value stream

Automate build and deployment processes

Automate unit tests and code analysis

Automate acceptance tests

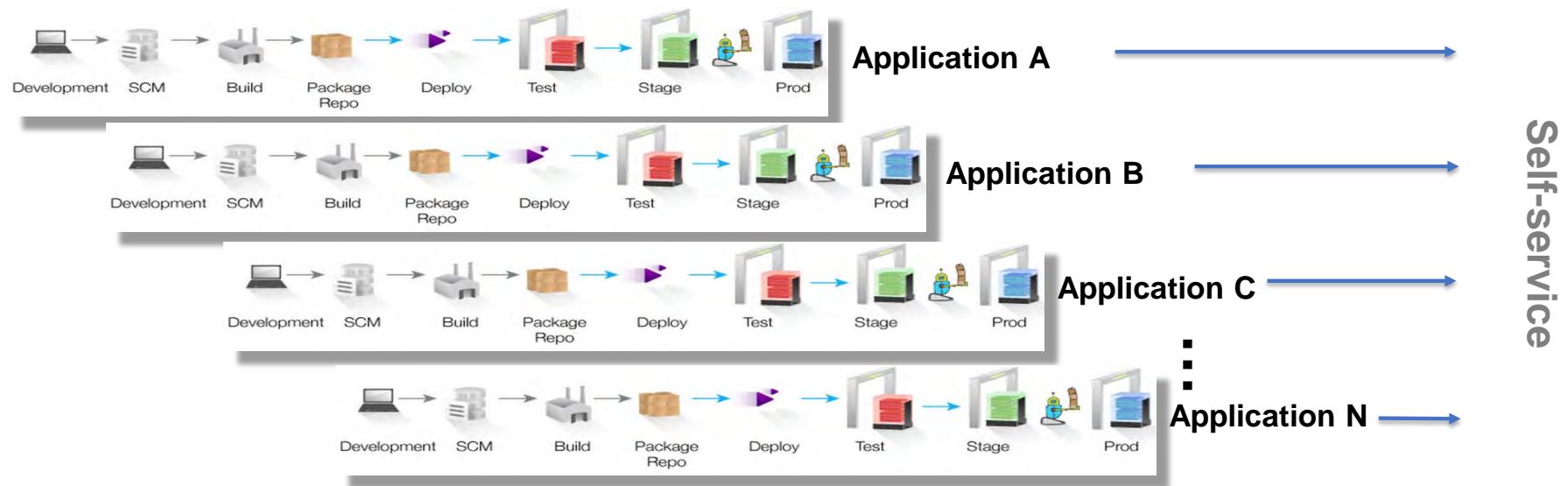
Ensure that each DevOps team member understands the capabilities and role of each tool in the DevOps toolchain to avoid tool overlap and toolchain functionality gaps.

Automate Releases

Add more automation as needed

Source: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and Dave Farley

# Multiple Business Applications Require Multiple Toolchains



Source: Sanjeev Sharma, IBM

Avoid creating more pipeline silos by taking an enterprise architecture approach: use 'sensible defaults'.

# Module 6: Quiz

- 1 DevOps is not about automation, just as astronomy is not about...
  - a) Telescopes
  - b) Stars
  - c) The sun
  - d) Spaceships
- 2 Who is responsible for the Periodic Table of DevOps Tools?
  - a) Chef
  - b) Google
  - c) IT Revolution
  - d) Xebia Labs
- 3 Fannie Mae reduced deployment time to 45 minutes from:
  - a) 3 hours
  - b) 30 hours
  - c) 3 days
  - d) 30 days
- 4 Who is best placed to provide DevOps toolchains as a shared service?
  - a) Development
  - b) The business
  - c) Infosec
  - d) IT Operations
- 5 Patrick Debois said: "Your tools alone will not make you..."
  - a) Profitable
  - b) Competitive
  - c) Successful
  - d) A market leader

# Module 6: Quiz Answers

- |   |  |   |
|---|--|---|
| 1 | DevOps is not about automation, just as astronomy is not about...    | a) Telescopes<br>b) Stars<br>c) The sun<br>d) Spaceships                      |
| 2 | Who is responsible for the Periodic Table of DevOps Tools?           | a) Chef<br>b) Google<br>c) IT Revolution<br><b>d) Xebia Labs</b>              |
| 3 | Fannie Mae reduced deployment time to 45 minutes from:               | a) 3 hours<br>b) 30 hours<br><b>c) 3 days</b><br>d) 30 days                   |
| 4 | Who is best placed to provide DevOps toolchains as a shared service? | a) Development<br>b) The business<br>c) Infosec<br><b>d) IT Operations</b>    |
| 5 | Patrick Debois said: "Your tools alone will not make you..."         | a) Profitable<br>b) Competitive<br><b>c) Successful</b><br>d) A market leader |

# Module 7

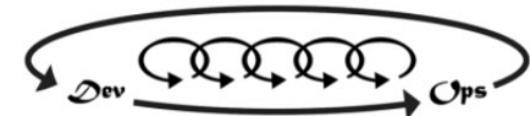
# **MEASUREMENT, METRICS & REPORTING**

# Module 7: Measurement, Metrics & Reporting

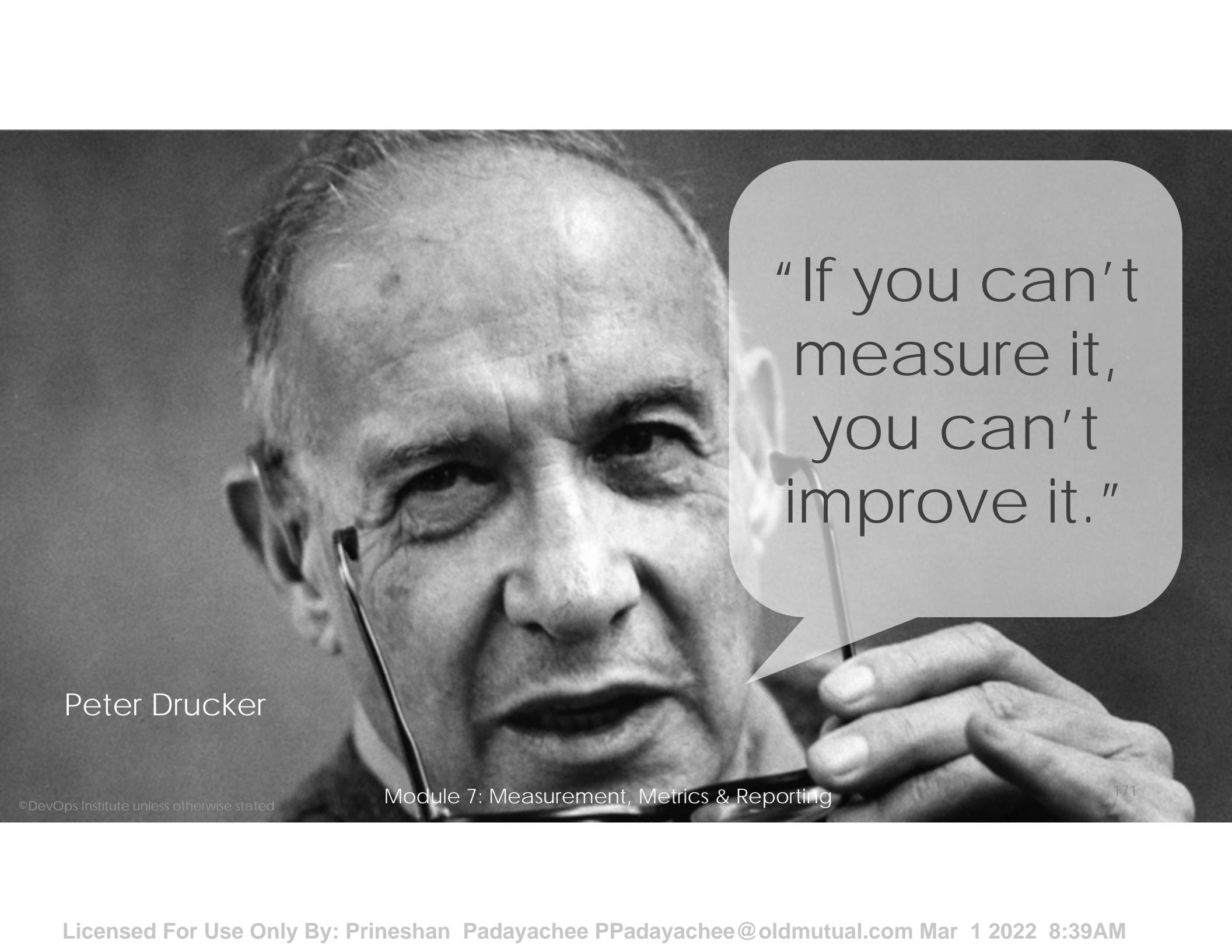
- The Importance of Measurement
- DevOps Metrics
  - Speed/Throughput/Tempo
  - Quality
  - Stability
  - Culture
- Change lead/cycle times
- Value Driven Metrics

Component	Module 7 Content
Video	4 DevOps Metrics to Improve Delivery Performance
Case Story	Societe Generale
Discussion	Metrics Used Today
Exercise	The Most Meaningful Metrics

# The Importance of Measurement



The First Way	The Second Way	The Third Way
Flow	Feedback	Continuous Experimentation & Learning
<ul style="list-style-type: none"><li>• Change lead time</li><li>• Change cycle time</li><li>• Time to value</li><li>• Value realisation</li></ul>	<ul style="list-style-type: none"><li>• Build/test results</li><li>• Change fail rate</li><li>• Monitoring</li><li>• % rework / complete &amp; accurate</li></ul>	<ul style="list-style-type: none"><li>• Hypothesis log</li><li>• Time allocated</li><li>• Time spent</li><li>• Mastery achieved and reported</li></ul>
Measurements allow us to find constraints and justify their removal and monitor improvement	Evidence builds trust and earns the right to do more – placing the bets in experimentation	Hypotheses need quantifiable outcomes to determine next experiment

A black and white close-up photograph of Peter Drucker's face. He has thinning hair and is looking slightly downwards and to his right with a thoughtful expression. A stethoscope hangs around his neck. A speech bubble originates from his mouth, containing his famous quote.

“If you can't measure it,  
you can't  
improve it.”

Peter Drucker

The video player displays a presentation slide with the following content:

- Software Development**: Lead Time
- Software Deployment**: Change Fail
- Service Operation**: Availability
- Four Key Metrics**: Deployment Frequency, Time to Restore

The slide is set against a background image of a woman speaking on stage at the Open Source Summit North America.

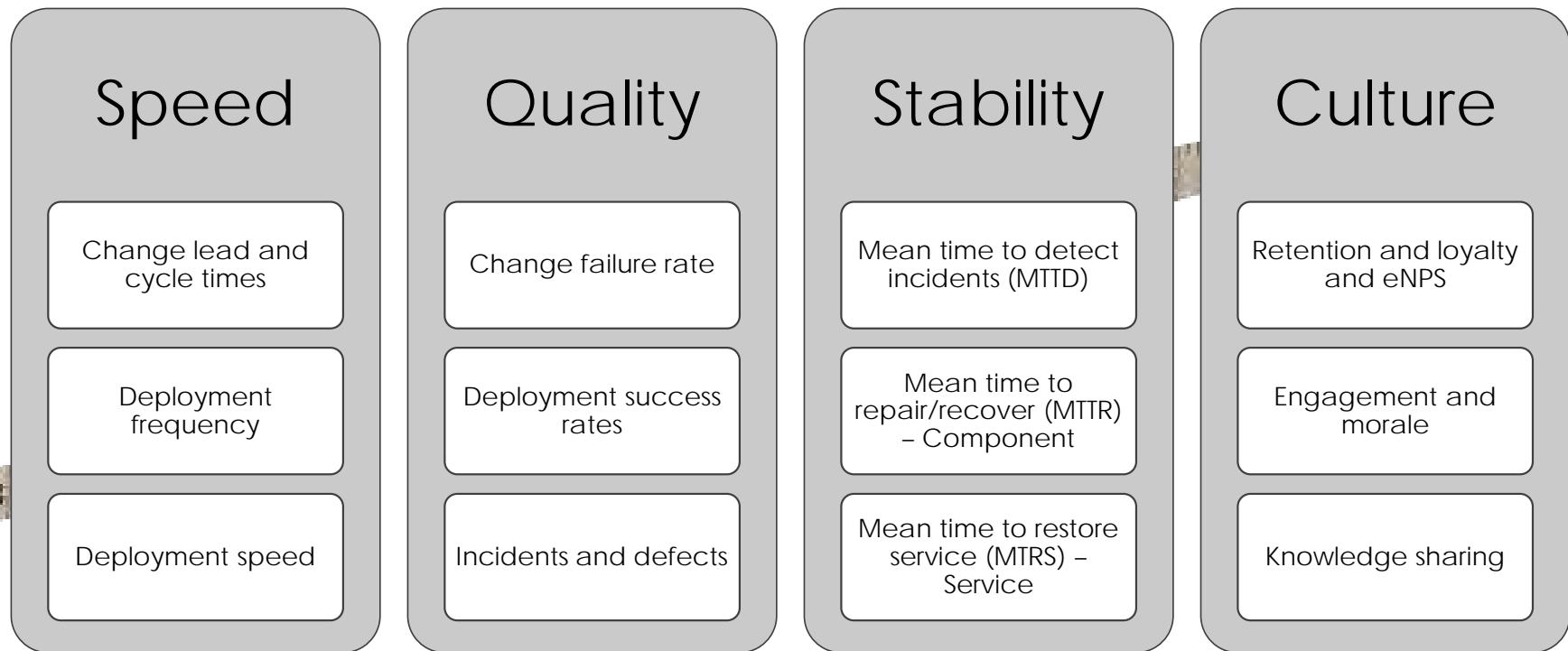
**Up next**

- Jez Humble - Building and Scaling High Performing...** Prezi Conference Team 1.2K views 44:58
- Visual Basic .Net : Search in Access Database -...** iBasskung Recommended for you 24:59
- GOTO 2015 • Agile is Dead • Pragmatic Dave Thomas** GOTO Conferences 704K views 40:39
- DebConf 14: QA with Linus Torvalds** Plent Sessi Adnan Hodzic



Double the Awesome  
with Dr. Nicole Forsgren (21:46)

# Measuring Success



Showing proof that DevOps practices benefit the organization requires examining factors that influence overall IT performance.

Adapted from  
Splunk 2016

## Module 7: MEASUREMENT, METRICS & REPORTING

# DISCUSSION

## Metrics Used Today

# Change Lead/Cycle Time

Lead Time	Cycle Time
The total elapsed time from the point when a user story enters the backlog, until the time it is completed – including the time spent waiting in a backlog.	The time it takes for a story to go from being “In progress” to Done.

Lead Time minus Cycle Time is Wait Time

Source: Accelerate: Dr Nicole Forsgren, Jez Humble & Gene Kim

# Guidelines to Measure IT Performance

DON'T MEASURE	DO MEASURE
Outputs, productivity	Outcomes, value
Maturity	Capability
Lines of code, velocity, utilization	Delivery lead time, deployment frequency, time to restore service, change fail rate
Individual or local	Team or global

4 types of (IT) work:

- Business projects
- IT projects
- Planned Work
- Unplanned work

Accelerate: Dr Nicole Forsgren, Jez Humble & Gene Kim

# CASE STORY: Societe Generale

"It's important to establish two sets of indicators. The first is the transformation itself. In other words, you need to measure how fast you're moving towards the transformation. The second indicator is about the business value - what is the time to market from idea to production, including sprint velocity and quality?"



Carlos Gonsalves, Global  
Chief Technology

"The return on investment (ROI) of the effort is extremely important for others in the organization."

## Benefits

- Transitioned from a high-workload, waterfall-based approach
- Turned around an unsatisfied user base
- Continuous Delivery has seen:
  - 45% reduction in time-to-market
  - 10% savings in their (very considerable) operating budget

# Gartner DevOps Metrics Pyramid



Module 7: Measurement, Metrics & Reporting

178

## Module 7: MEASUREMENT, METRICS & REPORTING

### EXERCISE

#### The Most Meaningful Metrics

# Module 7: Quiz

- |   |  |  |
|---|--|--|
| 1 | What does evidence build?  | a) Pain<br>b) Suspicion<br>c) Trust<br>d) Speed  |
| 2 | The research to which reports does the book 'Accelerate' describe in detail? | a) The Annual DevSecOps Reports<br>b) The State of DevOps Reports<br>c) The Best Jobs in America Reports<br>d) Most Popular DevOps Tools Reports |
| 3 | Peter Drucker said: "If you can't measure it, you can't..."                  | a) See it<br>b) Count it<br>c) Improve it<br>d) Feel it  |
| 4 | Which of these should you measure?   | a) Maturity<br>b) Capability<br>c) Outputs<br>d) Lines of code   |
| 5 | How do you calculate wait time?  | a) Lead Time minus Cycle Time<br>b) Cycle Time minus Lead Time<br>c) Cycle Rate minus Velocity<br>d) Velocity divided by Cycle Time              |

# Module 7: Quiz Answers

- |   |  |   |
|---|--|---|
| 1 | What does evidence build?  | a) Pain<br>b) Suspicion<br><b>c) Trust</b><br>d) Speed  |
| 2 | The research to which reports does the book 'Accelerate' describe in detail? | a) The Annual DevSecOps Reports<br><b>b) The State of DevOps Reports</b><br>c) The Best Jobs in America Reports<br>d) Most Popular DevOps Tools Reports |
| 3 | Peter Drucker said: "If you can't measure it, you can't..."                  | a) See it<br>b) Count it<br><b>c) Improve it</b><br>d) Feel it  |
| 4 | Which of these should you measure?   | a) Maturity<br><b>b) Capability</b><br>c) Outputs<br>d) Lines of code   |
| 5 | How do you calculate wait time?  | <b>a) Lead Time minus Cycle Time</b><br>b) Cycle Time minus Lead Time<br>c) Cycle Rate minus Velocity<br>d) Velocity divided by Cycle Time              |

# Module 8

## SHARING, SHADOWING & EVOLVING

# Module 8: Sharing, Shadowing & Evolving

- DevOps Days
- DevOps in the Enterprise
- Roles
- DevOps Leadership
- Organizational Considerations
- Getting Started
- Challenges, Risks and Critical Success Factors

Component	Module 8 Content
Video	DevOps: A Culture of Sharing
Case Story	Disney
Discussion	What's your open space topic?
Exercise	Write your personal action plan as an experiment

# DevOps Encourages a Sharing Culture

- Immersion opportunities are becoming more available in an effort to provide DevOps teams access to subject matter coaches on topics such as CI, CD, Lean and design methods
  - Dojos (Internal to Target)
  - Garages (IBM)
  - Lofts (Amazon)
  - More to come
- DevOps simulations and gamifications are also becoming more available

<http://target.github.io/devops/the-dojo>

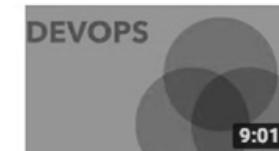
Games, hackathons, common workspaces, simulations and other innovations are helping to encourage the sharing of tools, knowledge, discoveries and lessons learned.

▶ (2) DevOps: A Culture of Sharing - YouTube - <https://www.youtube.com/watch?v=8aJHtp--3U>

▶ youtube.com



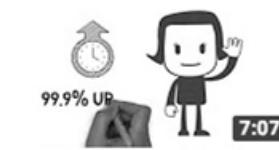
Up next



What is DevOps?  
CBT Nuggets  
98K views



The Culture Change of DevOps  
John Okoro  
2.8K views



What is DevOps? - In Simple English  
Rackspace  
891K views



Charlie Munger Interview 2018

# DevOps: A Culture of Sharing with Gareth Rushgrove (2:19)



Puppet  
Published on Apr 25, 2016  
©DevOps Institute. All rights reserved.

Module 8: Sharing, Shadowing & Evolving

SUBSCRIBE 12K

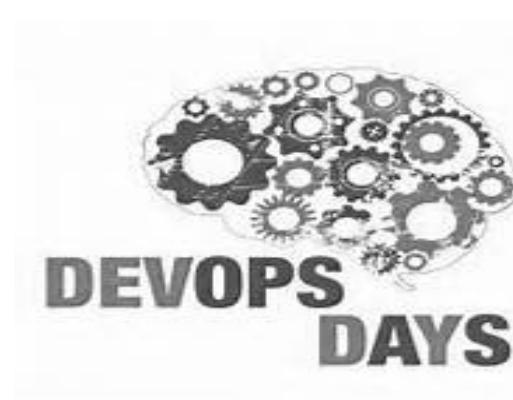


Recommended for you

# Internal DevOps Days

- Some organizations are replicating the DevOps Days model as internal events
- DevOps Days events give teams and individuals an opportunity to learn, share, discuss, engage and provide input and feedback

While most effective in a physical location, internal DevOps Days can be conducted in a virtual environment.



The format can include

- Traditional 30-minute presentations from internal and external resources
- Ignite (5 minute rapid-fire) topic-specific sessions
- Open Space break-out discussions on suggested topics

## DISCUSSION

What's your Open Space topic?

# DevOps in the Enterprise

Module 8: Sharing, Shadowing & Evolving

©DevOps Institute unless otherwise stated

# CASE STORY: Disney

"There's no secret to creating digital magic. We keep moving forward, opening up new doors, doing more things because we're curious."

"The digital expansion of business means more work and firefighting."



Jason Cox,  
Director of Systems  
Engineering

## Benefits

- 30 minutes to update 100 servers instead of 8 hours
- Less system drift
- Delivering continually and consistently
- Halved the cost whilst delivering more (movies)

# Roles

Module 8: Sharing, Shadowing & Evolving

©DevOps Institute unless otherwise stated

# Addressing the DevOps Skills Gap

- The demand for DevOps resources is making it difficult for organizations to attract and retain talent
- The breakneck pace at which technologies are evolving is making it difficult for individuals to maintain a current skill set
- Ensuring individuals have the needed soft skills and are a good cultural fit adds to the hiring challenge

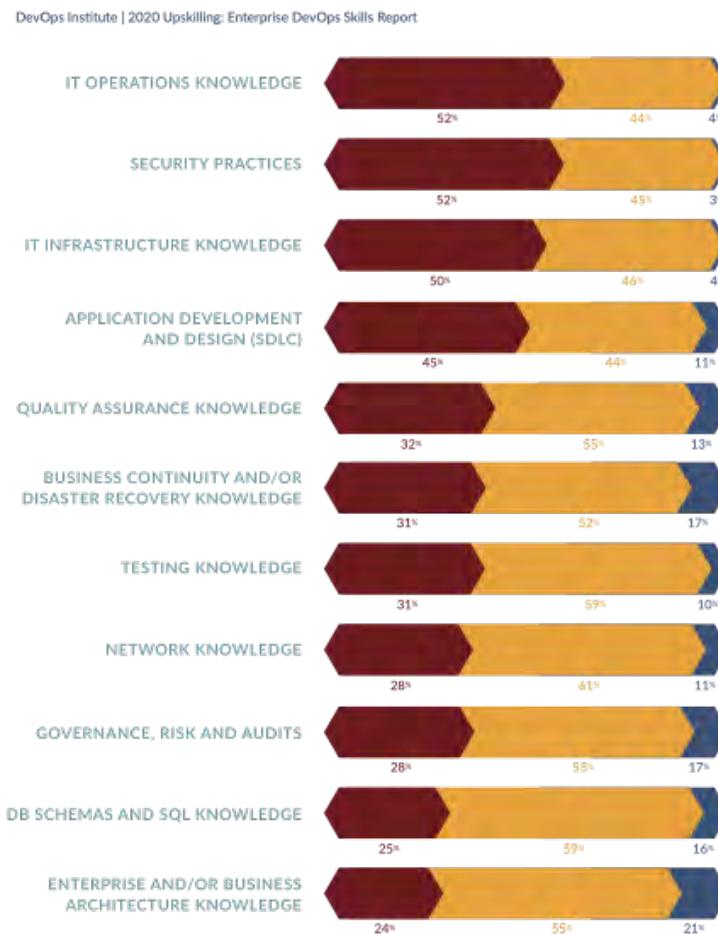
## Strategies

- Training and certification
- Immersion/coaching programs
- Restructuring pay and corporate culture
- Supplement internal teams with outsourced talent
- Recruiting bonuses

Today's CIOs are looking for workers who can shift gears and adapt to changing technology.

# IT Ops, Security and Infrastructure Skills

The DevOps human must have functional skills which include IT Operations (ITOps), security practices, IT infrastructure, application development and design (AD&D), quality assurance and business continuity/ disaster recovery (BC/DR)



**IT Ops, Security And Infrastructure Skills Are Just As Important As Application Development**

Figure 11: Functional Skills

Q How would you rate the importance of the following functional skills for your DevOps team members?

**Delta from 2019:**  
IT Infrastructure, network, security, GRC, BC/DR and Application Development all gained must-have votes from 2019.

- Very Important (Must-Have Skills)
- Important (Nice-to-Have Skills)
- Not Important (Optional Skills)

# Skills and Characteristics of a DevOps Professional

## Skills

- Business – Knowledge of business priorities and processes
- Technical – Specialist with broad generalist knowledge (T-shaped) – experience or at least an interest in writing code
- Soft – Communication, collaboration, team work
- Self-management – initiative, time and stress management, self-motivation, focus

## Characteristics

- Adaptable
- Customer-focused
- Craftsmen
- Curious
- Data-driven
- Engaged
- Empathetic
- Transparent

Generalist technical knowledge includes an understanding of DevOps practices, modern software engineering practices and modern architectures.

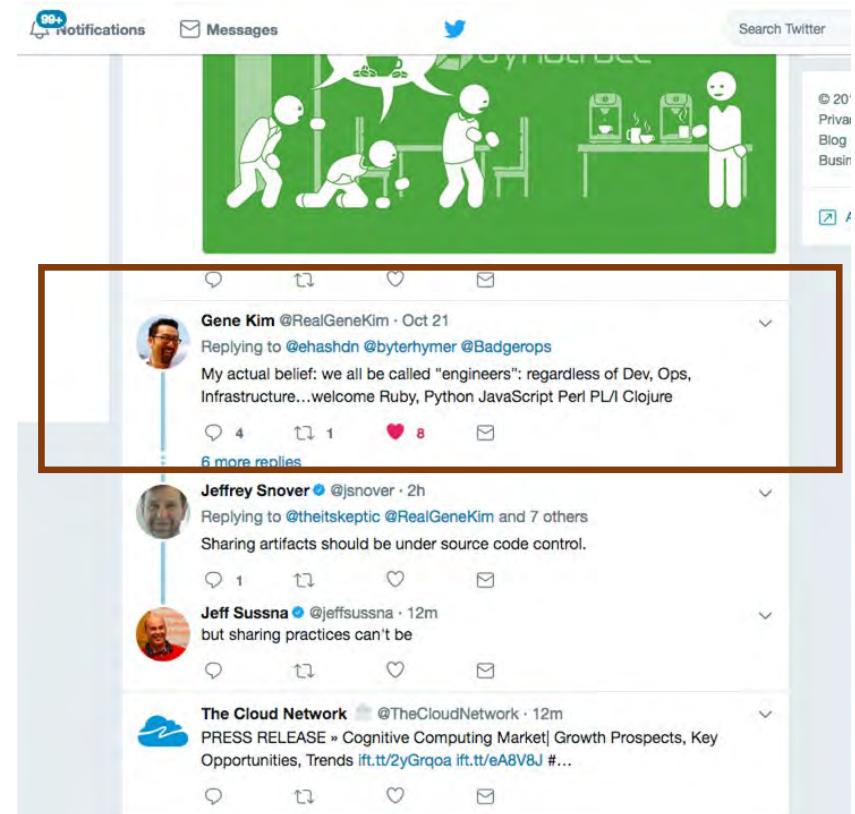
# DevOps Roles

What other roles do you think should be involved?

- DevOps evangelist or leader
- Software engineers, developers and testers
- Release manager
- Environment manager
- Product Owner
- Scrum Master
- Automation/continuous delivery architect
- Build engineer
- Security engineer
- Quality assurance (QA)/Experience assurance (XA)
- DevOps operations engineer
- IT Support
- Site Reliability Engineer
- Agile Service Manager®
- Agile Process Owner ®

# What is a DevOps Engineer?

- There is currently no 'industry recognized' job description or formal career track for a DevOps Engineer
- As with the concept of a DevOps team, the title has its pros and cons
- General characteristics include someone who
  - Wants to contribute his or her technical talent to business and process improvement initiatives
  - Is comfortable collaborating with others
  - Wants to be in a workplace that promotes a shared culture



# DevOps Leadership

©DevOps Institute unless otherwise stated

Module 8: Sharing, Shadowing & Evolving

# Transformational Leadership

"The goal of leadership is not to command, control, berate, intimidate, and evaluate workers through some set of contrived metrics. Instead, the job of leaders is to help organizations become better at self-diagnosis, self-improvement, and to make sure that local discoveries can be translated and converted to global improvements."

Dr Stephen Spear cited by Gene Kim in Beyond the Phoenix Project

## Dimensions of transformational leadership



The characteristics of transformational leadership are highly correlated with IT performance and employee Net Promoter Score (eNPS).

From The State of DevOps Report 2017

197

Module 8: Sharing, Shadowing & Evolving

©DevOps Institute unless otherwise stated

# Leading a Digital Transformation: According to Jason Cox (Disney)

Crucial Ingredients	Leadership Challenges
<ol style="list-style-type: none"><li>1. Collaboration - break down silos, mutual objectives</li><li>2. Curiosity - keep experimenting</li><li>3. Courage - candor, challenge, no blaming or witch-hunting</li></ol>	<ul style="list-style-type: none"><li>• The politics of command and control</li><li>• How new leadership can take a company in a new direction</li><li>• The blame bias of who versus what</li></ul>

# Organizational Considerations

Module 8: Sharing, Shadowing & Evolving

©DevOps Institute unless otherwise stated

# DevOps Organizational Structures

Some organizations are



- Assigning Ops liaisons to Dev/Scrum teams
- Creating cross-functional product (vs. project) teams
- Adopting matrix or market-oriented (vs. function-oriented) structures
- Creating shared Ops services that support multiple Dev teams



There is debate about the pros and cons of DevOps teams.

# DevOps Teams (1)

The creation of DevOps departments or teams was a growing trend; 16% in 2014, 19% in 2015, and 22% in 2016, 27% of 2017/2018 and 26% 2019 State of DevOps survey respondents indicated they were part of a DevOps department.

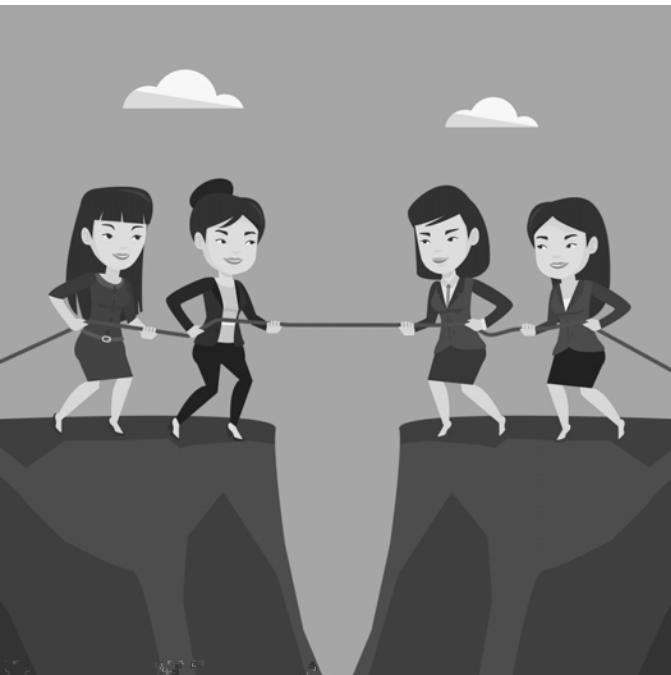
## DevOps teams:

- Expand upon the concept of an Agile or Scrum team
- Embed Dev and Ops skills into a single holistic group
- May be temporary or dedicated to a specific product
- May be cross-functional 'tiger teams' for short-term projects
- May evolve to provide shared services
- Have shared accountabilities
- Should adhere to the defined standards for development, automation, risk and compliance that applies to all DevOps teams

There is no 'ideal' structure for a DevOps team.



# DevOps Teams (2)



## Downsides of dedicated DevOps Teams:

- Less engagement across the IT value stream
- Risk of being another silo
- Dev and Ops wash their hands of accountability
- DevOps activities become someone else's problem

Regardless of structure, a DevOps team should be flat, with continuous engagement and the right balance of people, practices and automation skills.

# Getting Started

"It's a journey, not a silver bullet, and leaders need to avoid getting caught in analysis paralysis. Start making the changes, get the wins and let the organization evolve."

Melissa Sargeant



Module 8 Sharing, Shadowing & Evolving

204

# Start Where You Are



"DevOps is not your why, not your co-workers' why, certainly not your business' why."

Damon Edwards

- Get clear on the business opportunity – the 'Why?'
- Get the right people together
- Get everyone on the same page
- Invest in training and skills development
- Build capabilities that lead to lasting change
- Focus on critical behaviors
- Experiment and learn
- Consolidate gains and produce more change
- Avoid inertia

=RUN  
DECK

Module 8: Sharing, Shadowing & Evolving

205

# Learn by Doing

- Create a pilot where you can maximize the probability of success
- It should be small enough where
  - Success is apparent and understood
  - Consequences of failure aren't so large that a mistake could shut down the entire initiative
- It should be large enough that
  - You can show proof of improvement
  - You earn the right to make future improvements

# Consolidate Gains and Produce More Change

- Communicate successes, failures and lessons learned
- Document and make available reusable artifacts and measurements
- Expand your cycles of improvement
- Continuously invest in education
- Introduce advanced tools and techniques as needed

# Anchor the Results

- Prove that the new way of doing things is better
- Reinforce new behaviors with incentives and rewards
- Be prepared to lose some people along the way
- Reinforce the new culture with every new employee

*“Change sticks when it becomes ‘the way we do things around here’.”*

John. P. Kotter

Module 8: SHARING, SHADOWING & EVOLVING

# Challenges, Risks & Critical Success Factors

# Critical Success Factors

- Management commitment to culture change
- Creation of a collaborative, learning culture
- Training and continuous skills improvement
- Common values and vocabulary
- Systems engineering that spans Dev and Ops
- Meaningful metrics
- A balance between automation and human interaction
- Application of agile and lean methods
- Open and frequent communication

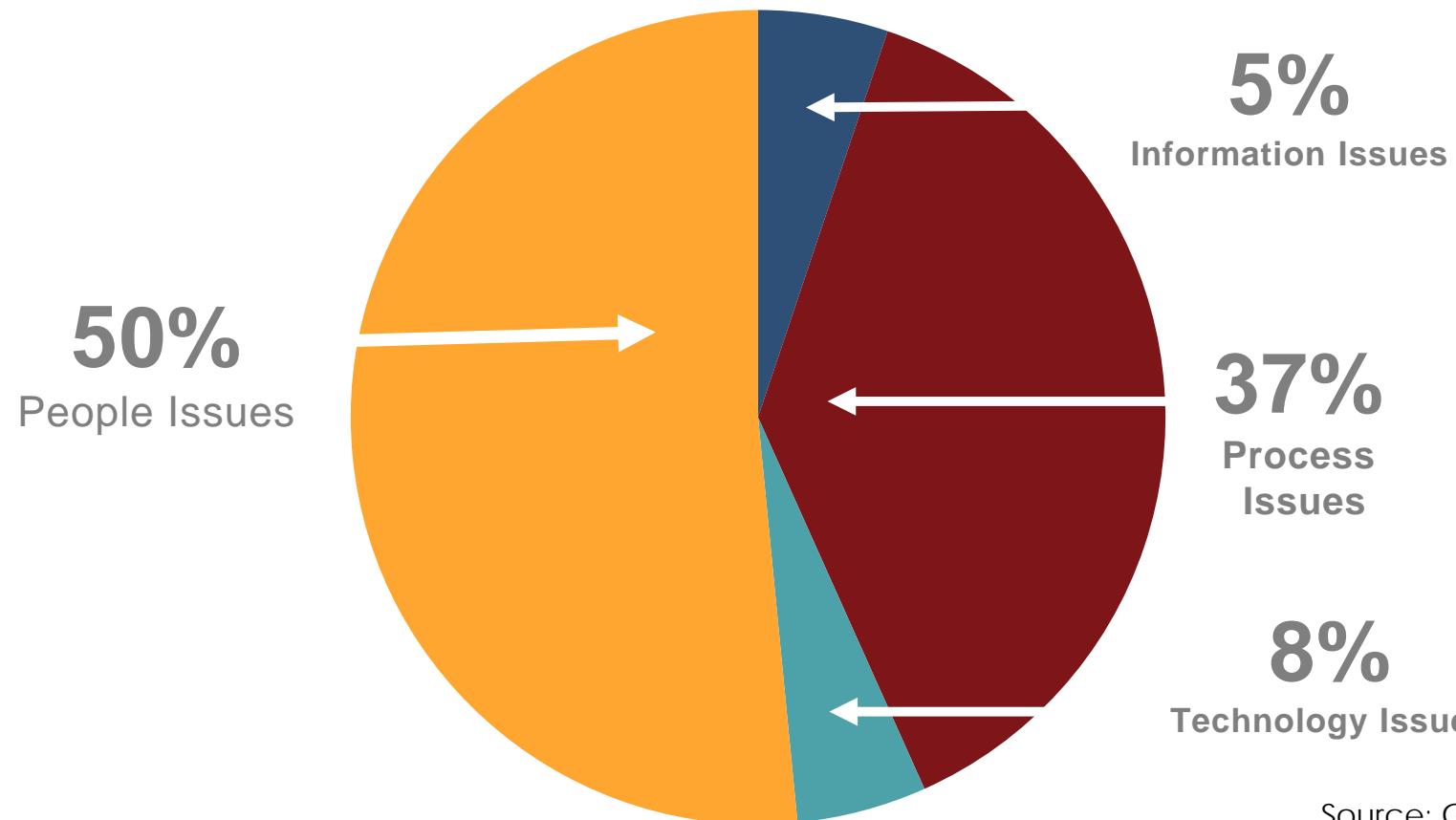
# Challenges and Risks

Overcoming these challenges will require organizational change.



- Lack of commitment or clarity
- Transforming a “them” and “us” culture
- Blending teams that are geographically dispersed, unfamiliar with each other and may include suppliers
- Lack of education, training and skill
- Immature service management processes
- Inadequate technologies
- Poor communication

# Your Biggest Challenge for the Expansion of DevOps?



Source: Gartner

Module 8: Sharing, Shadowing & Evolving

212

## Module 8: SHARING, SHADOWING & EVOLVING

# EXERCISE Your DevOps Experiment (Personal Action Plan)

# Module 8: Quiz

- |   |   |   |
|---|---|---|
| 1 | Who provides 'lofts' as an immersive experience?  | a) Amazon<br>b) IBM<br>c) Google<br>d) Puppet   |
| 2 | What's an 'Ignite' in the context of DevOps Days?   | a) A key note<br>b) A 30 minute presentation<br>c) A 5 minute topic specific session<br>d) A breakout session |
| 3 | Why does Disney keep doing more things?<br>Because they're:   | a) Impatient<br>b) Competitive<br>c) Curious<br>d) Courageous   |
| 4 | What's the ideal structure for a DevOps team?   | a) Matrix<br>b) Market oriented<br>c) There isn't one<br>d) Cross functional                                  |
| 5 | According to Gartner, what is the most common challenge for organizations adopting DevOps principles? | a) People<br>b) Information<br>c) Process<br>d) Technology  |

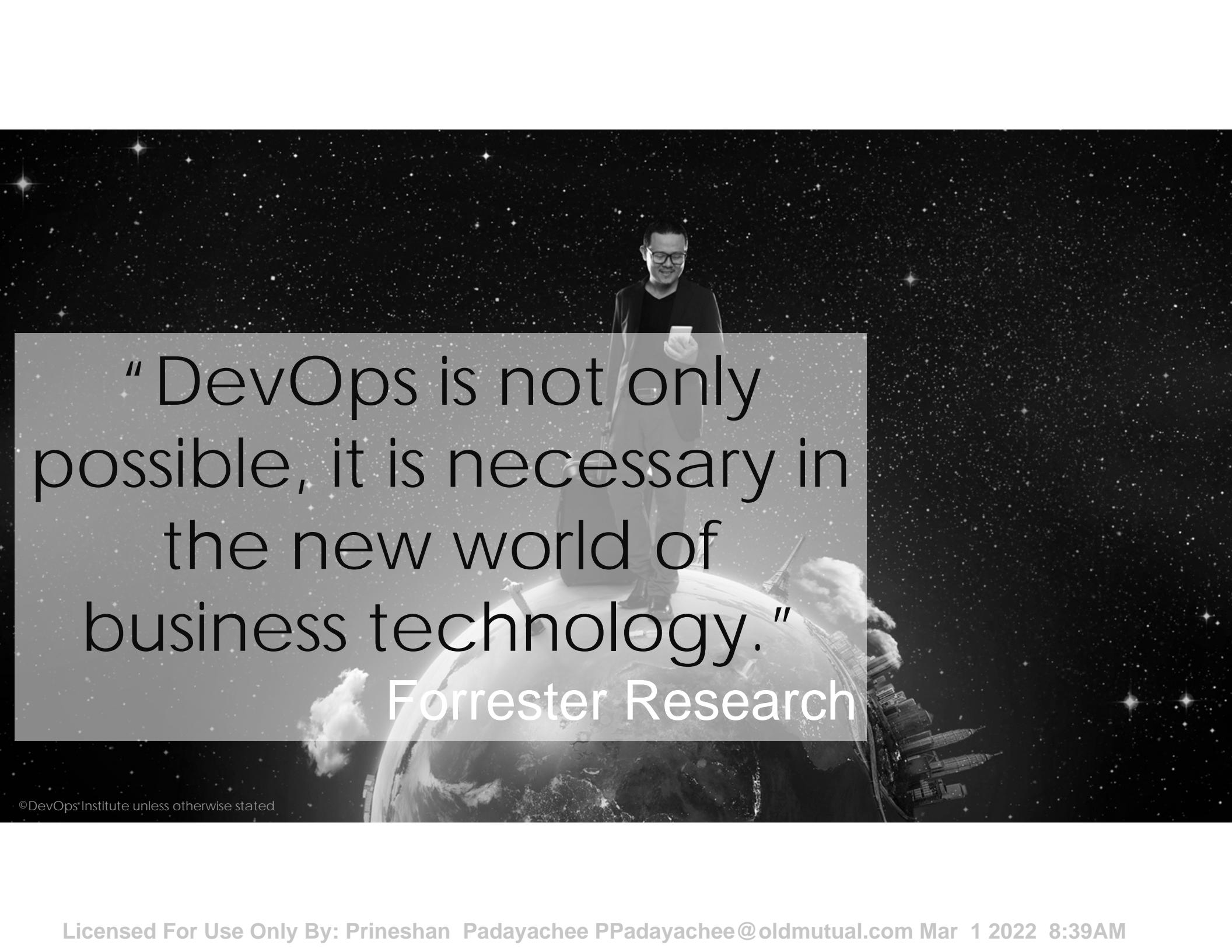
# Module 8: Quiz Answers

- |   |   |  |
|---|---|--|
| 1 | Who provides 'lofts' as an immersive experience?  | a) Amazon<br>b) IBM<br>c) Google<br>d) Puppet  |
| 2 | What's an 'Ignite' in the context of DevOps Days?   | a) A key note<br>b) A 30 minute presentation<br><b>c) A 5 minute topic specific session</b><br>d) A breakout session |
| 3 | Why does Disney keep doing more things?<br>Because they're:   | a) Impatient<br>b) Competitive<br><b>c) Curious</b><br>d) Courageous   |
| 4 | What's the ideal structure for a DevOps team?   | a) Matrix<br>b) Market oriented<br><b>c) There isn't one</b><br>d) Cross functional                                  |
| 5 | According to Gartner, what is the most common challenge for organizations adopting DevOps principles? | <b>a) People</b><br>b) Information<br>c) Process<br>d) Technology  |

# Summary

DevOps enables companies to deliver better software faster and more reliably by...

- Improving communication, collaboration and the integration of processes and tools across the IT value stream
- Automating the process of software delivery and infrastructure changes
- Leveraging agile, lean, ITSM and evolving DevOps practices

A black and white photograph of a man with glasses and a suit, standing on a stylized Earth globe. He is pointing upwards with his right hand. The background is a dark, star-filled space.

“DevOps is not only  
possible, it is necessary in  
the new world of  
business technology.”

Forrester Research

©DevOps Institute unless otherwise stated

# Become Part of the DevOps Community



## Join The DevOps Member Association

Advancing the Human Elements of DevOps



# DevOps Institute Learning Tracks & Community

