



DevOps Foundation

Learner Manual

IMPORTANT: This is copyrighted material and access is authorized for the sole use of a single candidate to undertake personal training in this subject. Improper use of this document may result in legal action.



DevOps Foundation Table of Contents

Course Introduction	1
Module 1 – Exploring DevOps	7
Module 2 - Core DevOps Principles	49
Module 3 – Key DevOps Practices.....	67
Module 4 – Business & Technology Frameworks	86
Module 5 – Culture, Behaviors & Operating Models.....	118
Module 6- Automation and Architecting DevOps Toolchains	143
Module 7 – Measurement, Metrics & Reporting	169
Module 8 – Sharing, Shadowing & Evolving	183
DevOps Foundation Examination Requirements	219
Sample Examination and Answer Key	223
Value Added Resources	234
DevOps Glossary.....	247

DEVOPS FOUNDATION

© DevOps Institute unless otherwise stated

1

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

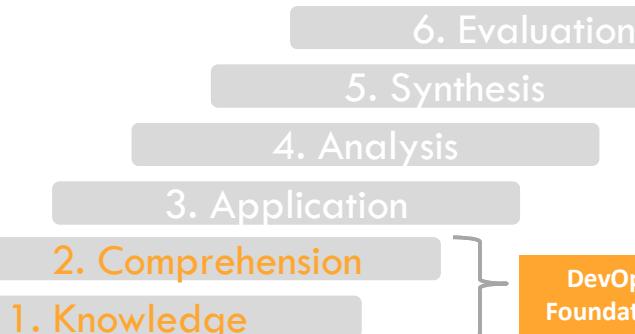
- Learn about DevOps
- Understand its core vocabulary, principles, practices and automation
- Hear and share real life scenarios
- Earn 16 PDUs (if applicable)
- Have fun!



Pass the DevOps Foundation Exam

- 40 multiple choice questions
- 60 minutes – 75 for English as a Second Language (ESOL) learners
- 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



The DevOps Institute (DOI) is the Continuous Learning Community for emerging DevOps practices.

DevOps Institute's vision is to facilitate a community where learning is a lifestyle and members have access to the most innovative, inspirational and transformational DevOps content, courses and certifications. We strive to provide content that inspires discussion, collaboration and transformation.

DevOps Foundation Course Content

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

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

The DevOps Collective Body of Knowledge

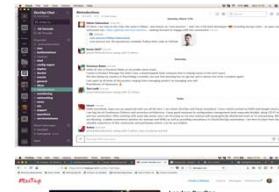
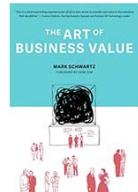
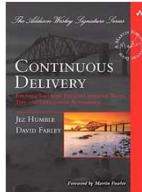
True to its core values, DevOps is emerging through a shared and collective body of knowledge (CBok) including:

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

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

Module 1: Exploring DevOps

The DevOps Collective Body of Knowledge (2)

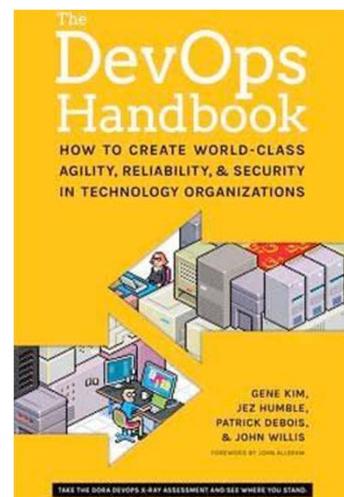


The DevOps Institute actively researches and influences emerging DevOps practices
in order to create meaningful training and certification.



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'*



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.

<http://devops.com/2016/03/17/what-devops-is-not/>

Why DevOps is Important Now

- Enterprises have young, nimble start-up 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

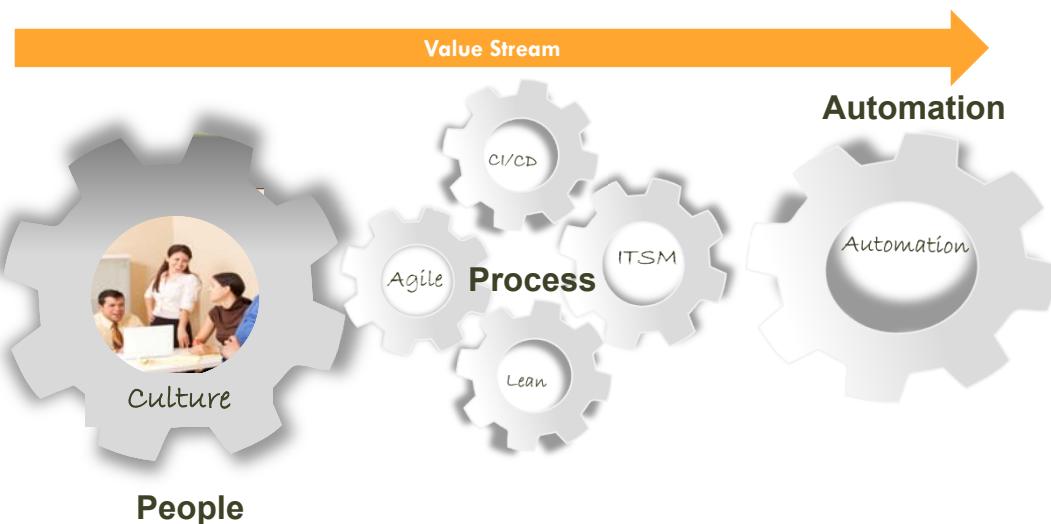
To meet these changing conditions, IT must adapt its culture, practices and automation to be more “continuous”.

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

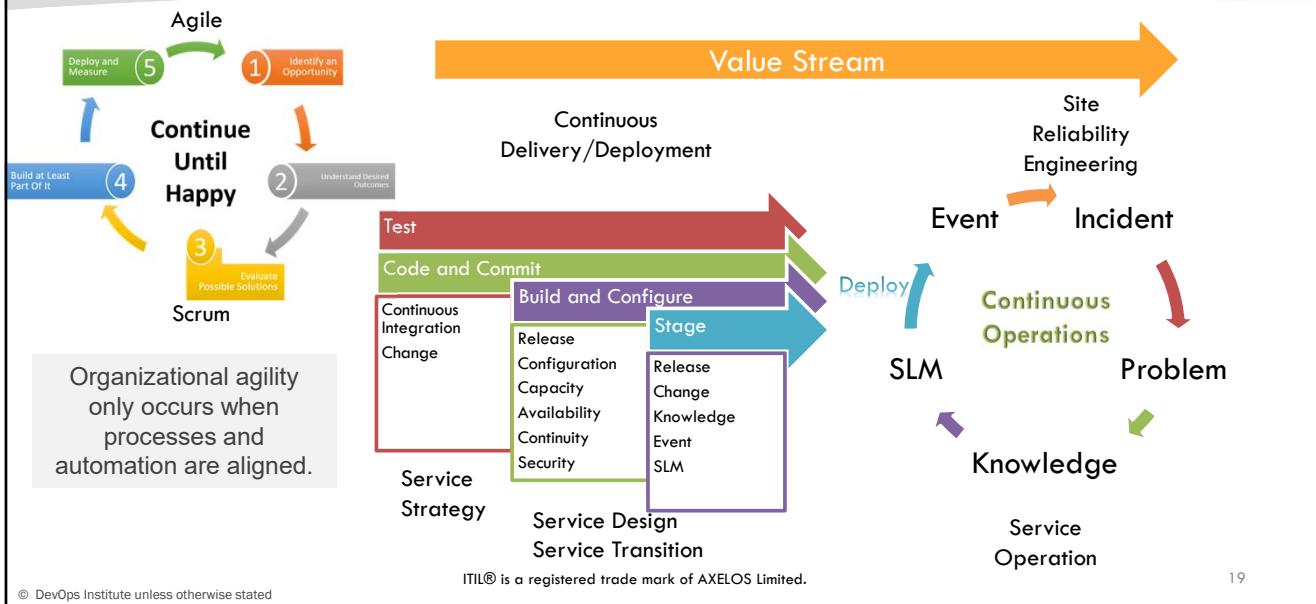


© DevOps Institute unless otherwise stated

ITIL® is a registered trade mark of AXELOS Limited.

18

IT is a System of Systems



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.



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.

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.

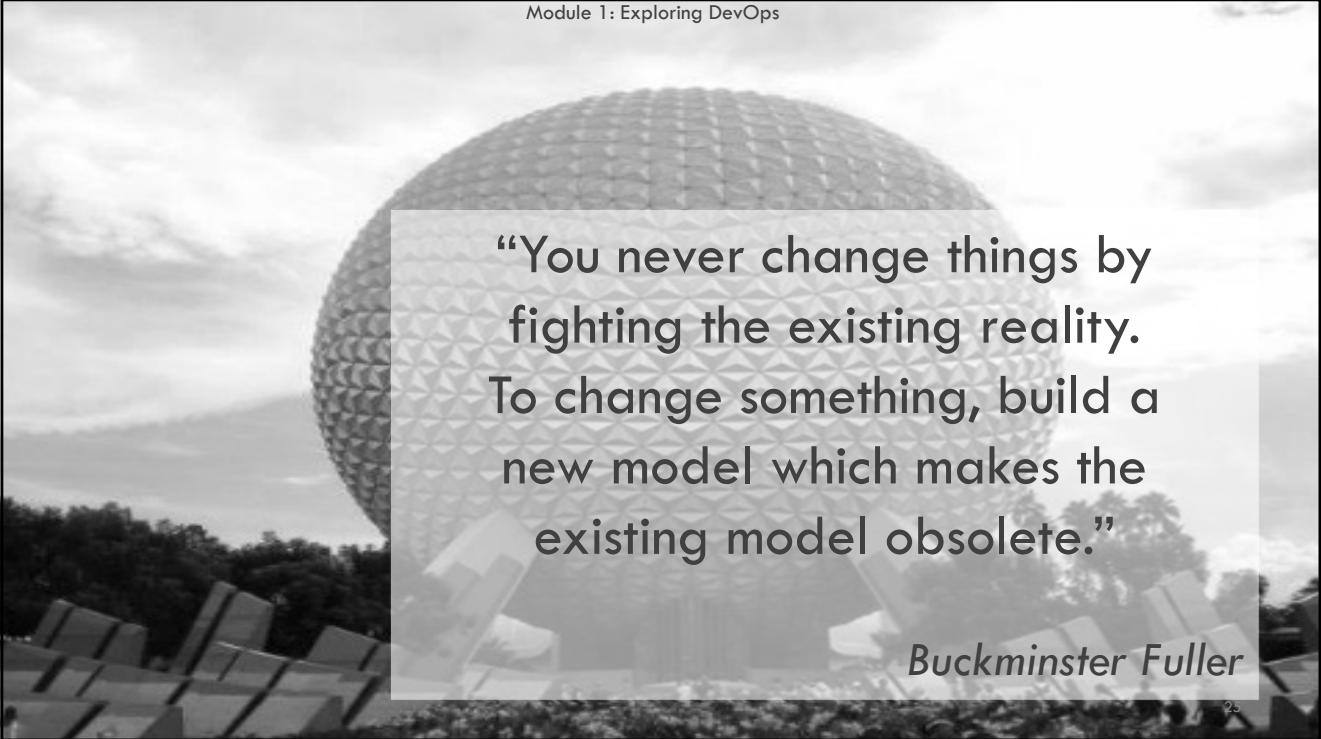
Ops includes all the people involved in delivering and managing software products and services including:

- 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.

24



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

Buckminster Fuller

Why Does DevOps Matter?

Our Cadence is Off

Historically...

The Business



Innovation



Dev



Waterfall Projects



Ops



Rigorous Processes



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 2018 State of DevOps Report, high-performing organizations have:

- 46 times more frequent code deployments
- 2555 times faster lead time from commit to deploy
- 2604 times faster time to recover from incidents
- 7 times lower change failure rate

Teams that reported the least transformative leaders were half as likely to be high performers. (2017)



"The Accelerate State of DevOps Report is the largest and longest-running research of its kind. It represents five years of work surveying over 30,000 technical professionals worldwide. The results allow us to better understand the practices that lead to higher software delivery performance that can generate powerful business outcomes. "

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

“IT has become
the beating
heart of the
bank.”

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



DevOps Adoption



GitLab 2018
Developer
Survey

- 65% of respondents say 'DevOps is a tremendous time saver'
- 81% of managers agree with the above statement
- 29% plan to invest in DevOps in 2018
- 23% say DevOps is their development methodology

FORRESTER® 2018: The Year Of Enterprise DevOps

- 13% have implemented
- 50% have implemented and are expanding
- 27% planning to implement (in the next 12 months)
- 9% interested but no immediate plans (in the next 12 months)
- 1% not interested

RIGHT SCALE® 2017 State of the Cloud Report

- Overall DevOps adoption continued to rise from 74 to 78 percent with enterprises reaching 84 percent
- 30 percent of enterprises are adopting DevOps company-wide, up from 21 percent in 2016

“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



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.

POWER C
POWER C
EMAIL
INTERNET
WORLD
NO
34
SEARCH

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

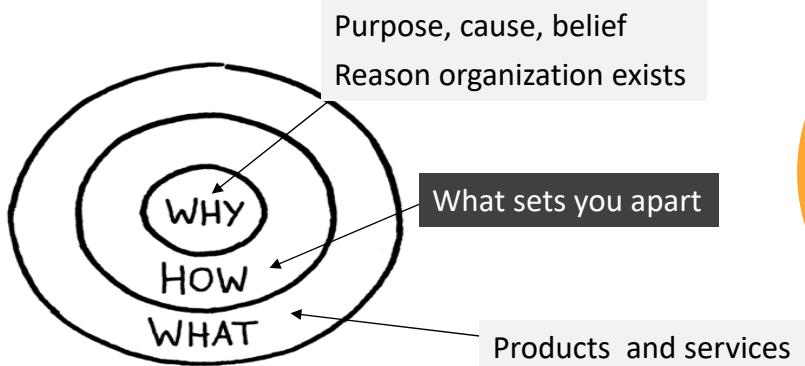


"Our analysis shows that elite performers are 1.53 times more likely to meet or exceed their goals for organizational performance, and high performers are 1.38 times more likely to meet or exceed their goals."

Source: 2018 State of DevOps Report

35

Start with the “Why” - The Golden Circle



© 2013 Simon Sinek, Inc.

36

© DevOps Institute unless otherwise stated

Module 1: Exploring DevOps



Start With Why
with Simon Sinek (5:00)

37

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**

© DevOps Institute unless otherwise stated



The IT Perspective

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?

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?



© DevOps Institute unless otherwise stated

43

The Wall of Confusion (2)



What does the business want?
All of the above

IT's Silo Culture



Dev

QA
Security



Ops

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.

Top 5 Predictors of IT Performance

DevOps helps IT organizations become high-performing.

1	2	3	4	5
Peer-reviewed change approval process	Version control for all production artifacts	Proactive monitoring	High-trust organizational culture	Win-win relationship between Dev & Ops

Source: 2014 State of DevOps Report – Puppet Labs, IT Revolution Press and ThoughtWorks

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) 25%
d) 10% |
| 5 | Who is responsible for The Golden Circle (organizational why)? | a) Simon Sinek
b) Alan Alter
c) Ron van Kermenade
d) Jez Humble |

© DevOps Institute unless otherwise stated

Module One Quiz Answers

- | | | |
|---|---|---|
| 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) 25%
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 |

© DevOps Institute unless otherwise stated

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

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)	Create a culture that fosters: <ul style="list-style-type: none"> Experimentation, taking risks and learning from failure Understanding that repetition and practice is the prerequisite to mastery

Module 2: Core DevOps Principles



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

53

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.

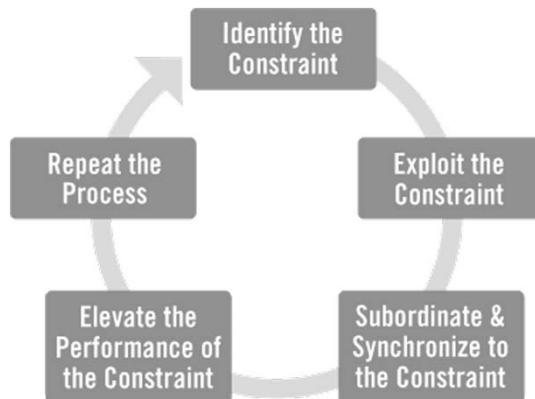
The Theory of Constraints recognizes that

- 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

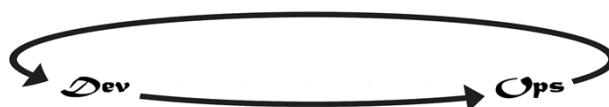
The Second Way: Feedback



- 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

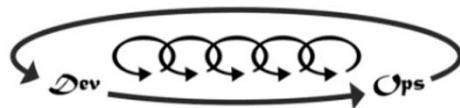
A goal of The Second Way is to shorten and amplify right to left feedback loops so necessary corrections can be continually made.

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.”



Justin Dean, VP
TechOps

ticketmaster

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

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

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 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 |

© DevOps Institute unless otherwise stated

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 |

© DevOps Institute unless otherwise stated

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."



Topo Pal,
Director & Platform
Engineering Fellow

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

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



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.

- Functional
 - Unit tests
 - API
 - Integration
 - System testing
- Non-functional
 - Performance
 - Security
 - Compliance
 - Capacity
- Test driven development (TDD)
- Performance testing
- End-to-end use-case testing
- Security testing
- Manual testing
- Canary & Blue/Green deployments

“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.

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

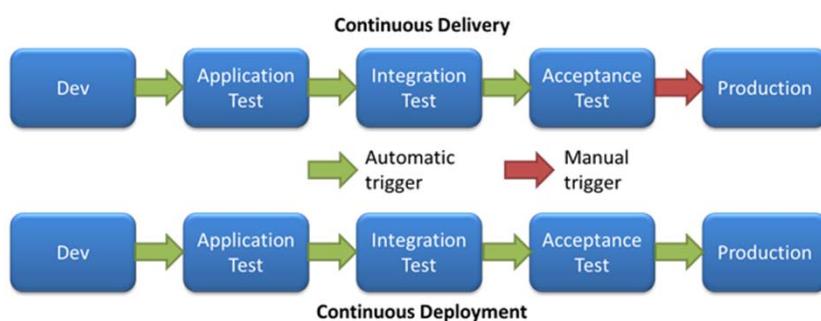
Factors that positively contribute to continuous delivery:



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,
author of 'DevOps for the Modern Enterprise'

75

Module 3: Key DevOps Practices



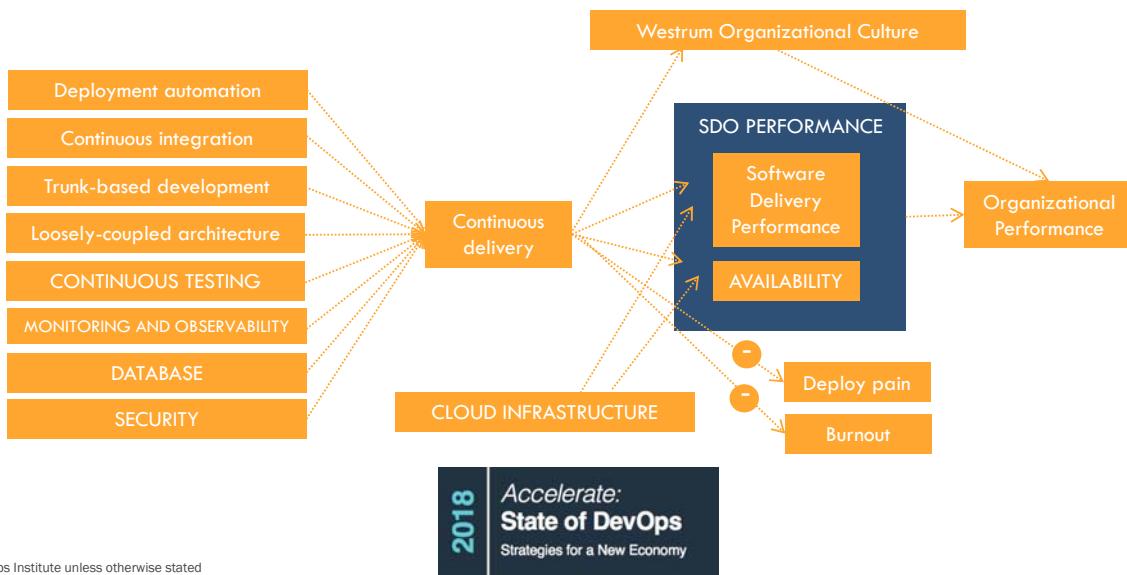
GitHub

Professional Guides



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

Continuous Delivery Can Lead to Higher IT & Business Performance

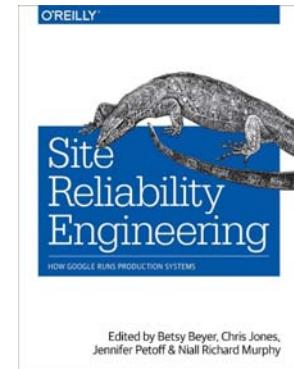


© DevOps Institute unless otherwise stated

77

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

78

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

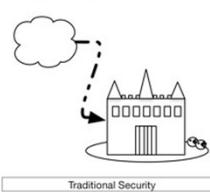
Who turned out the lights?

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.



Traditional Security

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...



Traditional Security



(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

© DevOps Institute unless otherwise stated

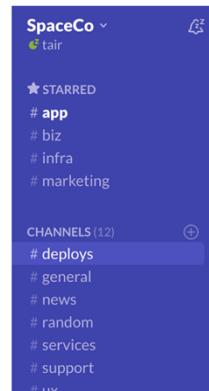
80

ChatOps

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



Google Hangouts



deploys
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 pro

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.

© DevOps Institute unless otherwise stated

81

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: 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 |

© DevOps Institute unless otherwise stated

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 |

© DevOps Institute unless otherwise stated

MODULE 4

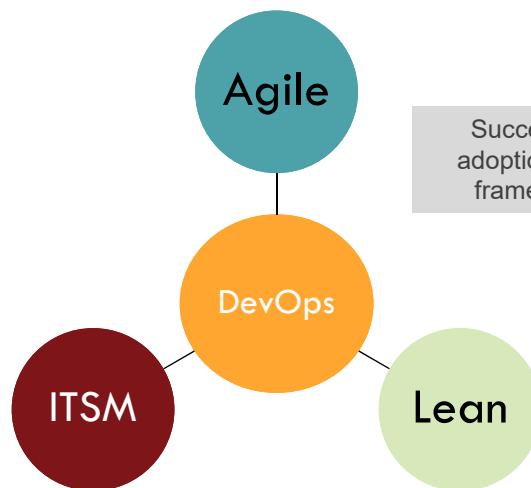
Business & Technology Frameworks

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

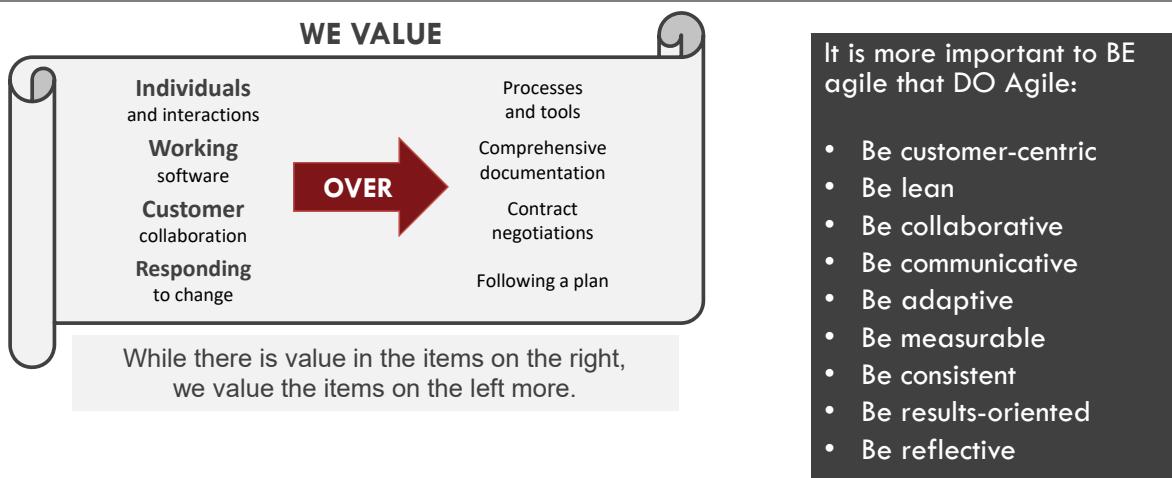


Successful DevOps relies on the adoption and integration of multiple frameworks and methodologies.

Agile

The Agile Manifesto

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



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.



Roles

- Product Owner
- ScrumMaster
- Development Team



Artifacts

- Product Backlog
- Sprint Backlog
- Increment

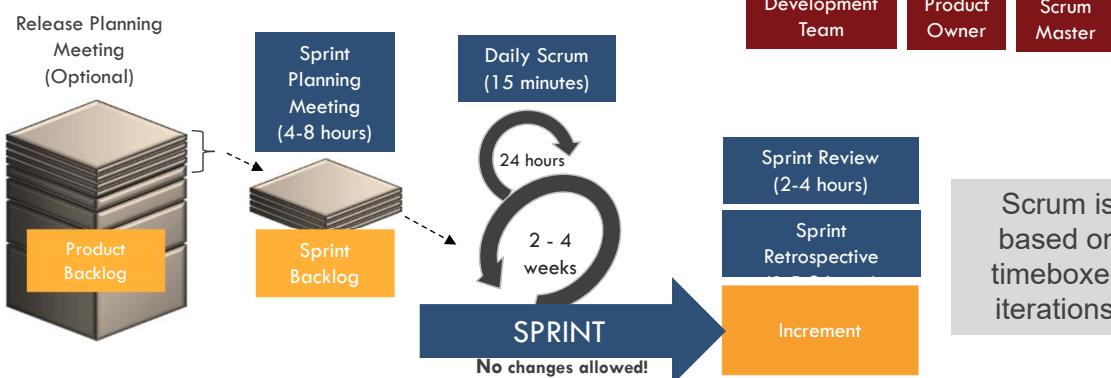


Meetings

- 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?

92

© DevOps Institute unless otherwise stated

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.

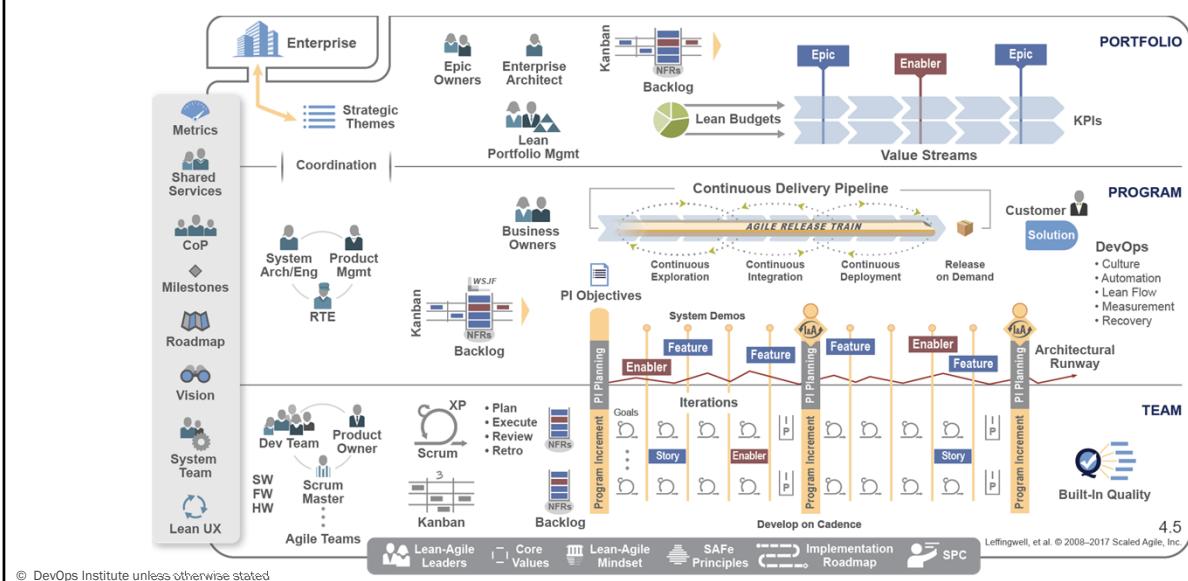
- Integrates Lean and Agile thinking into software development
- Focuses on iterative and incremental development, agile SW development, product development flow, lean thinking and field experience at enterprise scale
- Can be applied to organizations with a large number of practitioners and teams



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

Module 4: Business & Technology Frameworks

SAFe for Lean Enterprises



Spotify Engineering Culture
Part 1 & 2
Husk, Litteras = 1pm

Cross-pollination > Standardization

Internal Open-source model

Alignment

Community > Structure

Focus on Motivation

Trust > Control

Agile at scale requires Trust & scale

Politics **Fear**

95



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

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.



IT Service Management (ITSM)



Gene Kim

“It is my firm belief that ITSM and the DevOps movement are not at odds. Quite to the contrary, they’re a perfect cultural match.”

98

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 is a set of best practice publications for IT service management.



ITIL Core consists of five books that provide best practice guidance for an integrated service management process approach

- ITIL® Service Strategy
- ITIL® Service Design
- ITIL® Service Transition
- ITIL® Service Operation
- ITIL® Continual Service Improvement

ITIL process improvement supports DevOps practices.

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

ITSM Process Models Support DevOps and Continuous Delivery

- Predefined procedures
 - Steps to be taken
 - Chronological order and dependencies
 - Responsibilities
 - Timescales and thresholds
 - Escalation procedures
- Define steps for handling specific types of transaction
- Ensure a defined path or timeline is followed
- Can be automated

Examples

- Change models
- Release models
- Test models
- Incident models
- Problem models
- Request models

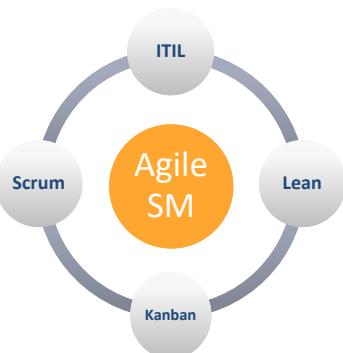
AXELOS®, ITIL® and IT Infrastructure Library® are registered trade marks of AXELOS Limited.
Based on ITIL Text - ST 4.2.4.5

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

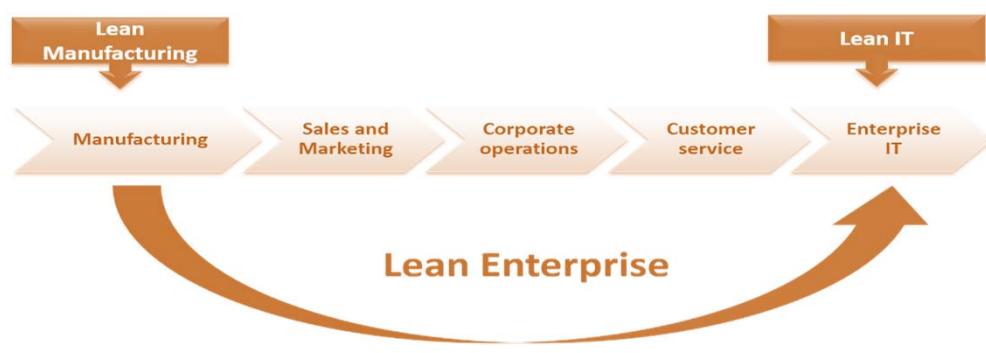
Source: Agile Service Management Guide

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

Lean

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.



Lean IT applies the key ideas behind lean production to the development and management of IT products and services.

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

Module 4: Business & Technology Frameworks



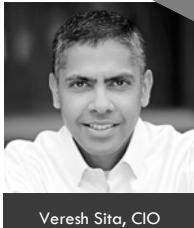
Gene Kim

“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.”

108

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



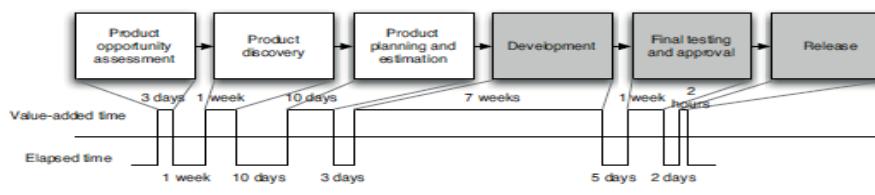
Value Stream Mapping

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.

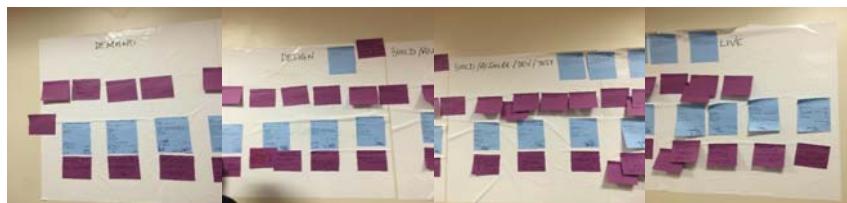
- A value stream is the sequence of activities required to design, produce, and deliver a specific product or service
- Value streams typically span multiple processes
- Value stream mapping enables cross-functional teams to
 - See an entire value stream from a work and information flow perspective
 - 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



Sample Value Stream Maps



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

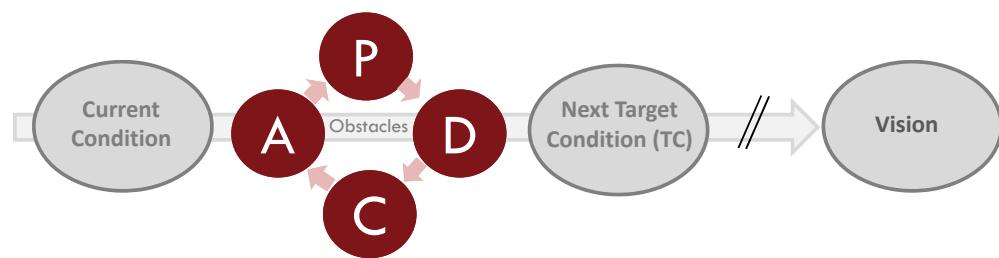


Source: Ranger4

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)

© DevOps Institute unless otherwise stated

Safety Culture

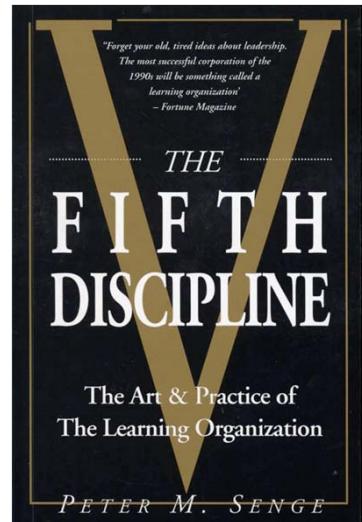
- Attitude, beliefs, perceptions and values that employees share in relation to safety in the workplace
- Blameless postmortems
- Valuing incidents
- Avoiding Single Points of Failures (SPOFs)
- The Andon Cord – thank you for creating a learning opportunity

“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



© DevOps Institute unless otherwise stated

114

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 |

© DevOps Institute unless otherwise stated

Module 4: Quiz Answers

- | | | |
|---|---|---|
| 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 |

© DevOps Institute unless otherwise stated

MODULE 5

Culture, Behaviors & Operating Models

Module 5: Culture, Behaviors & Operating Models

- Defining Culture
- Behavioral Models
- Organizational Models
- Target Operating 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

© DevOps Institute unless otherwise stated

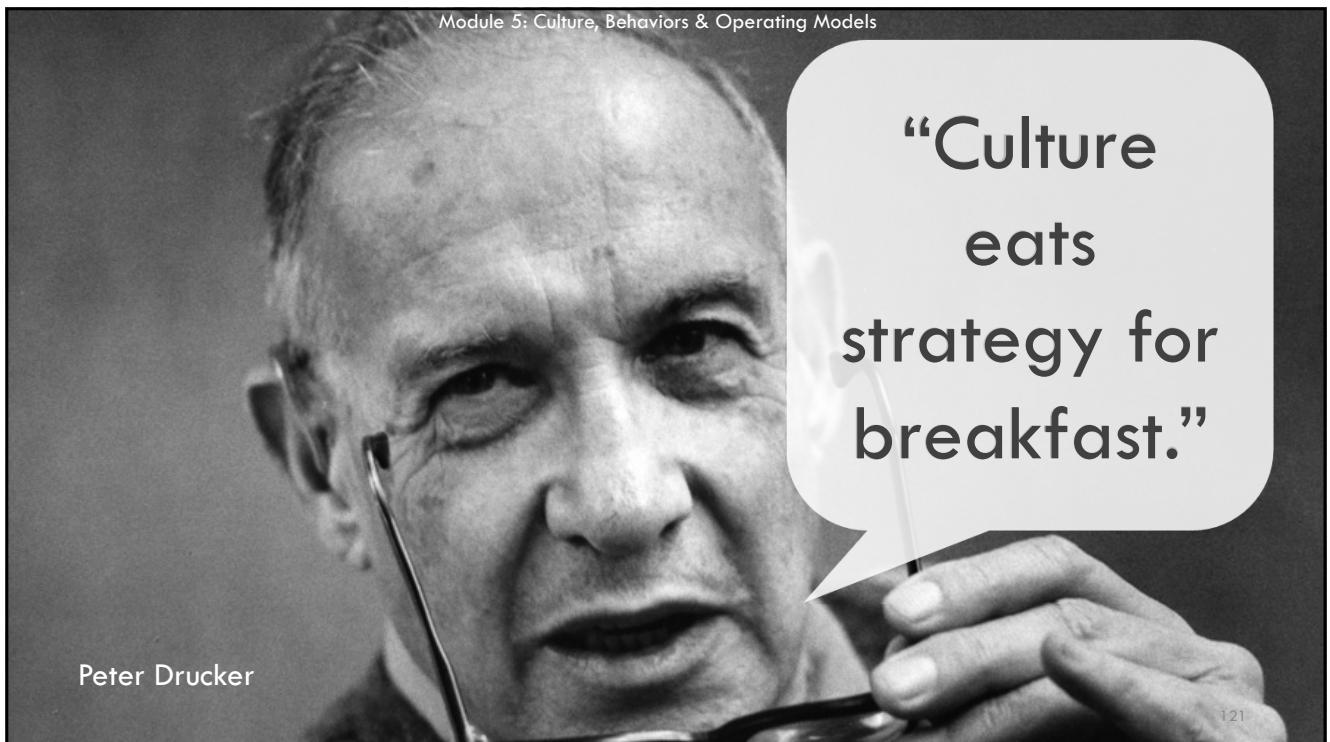
Practices Alone Are Not Enough

“Tools and processes are a reflection of your cultural choices.”

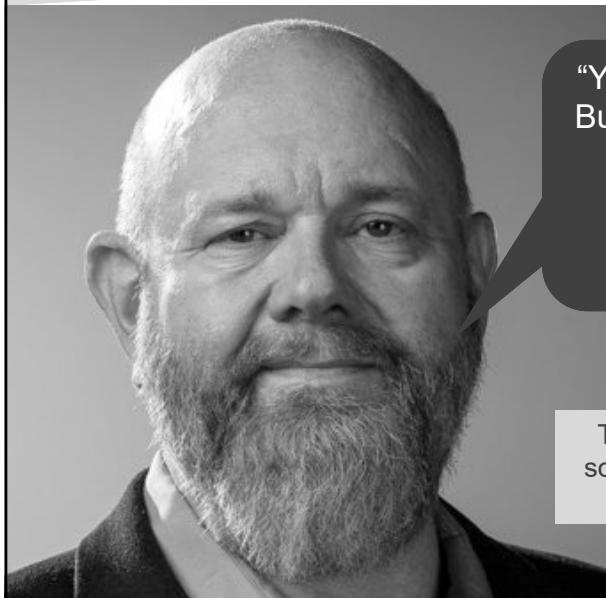
Sascha Bates



120



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

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 **HubSpot**

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!

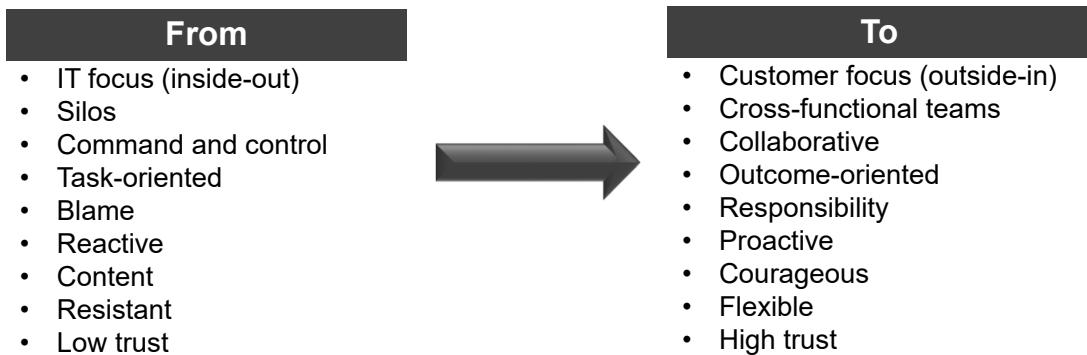
Characteristics of a DevOps Culture

- 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.

124

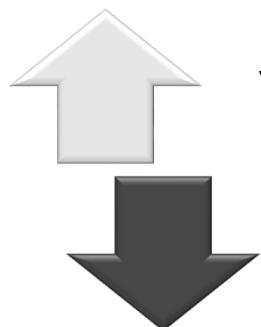
Shifting Thoughts and Behaviors



Real culture change takes time. It must be incremental and performed at a realistic pace.

High Trust vs. Low Trust

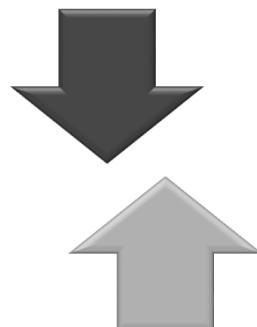
High Trust



Speed

Cost

Low Trust



Speed

Cost

Spotify Engineering Culture Part 2

Fail Fast → Learn Fast → Improve Fast

Failing Reasons > Failure Aesthetic

Limited Blast Radius
via Design Architecture
via Crashed Robot

Experiment-friendly Culture

Waste-repellent Culture → Lean

Continuous Improvement
Driven from within, supported from above.

Impact > Velocity

Impact AB Test

Impact > Predicted

Impact > Actual

Impact > Control

You are the culture
Model the behavior you want to see

Lean Startup

Idea/problem

Narrative → Prototypes

Build MVP

Test → **Release**

Innovation > Predictability
 $100\% \text{ predictability} = 0\% \text{ innovation}$

Focus on

Focus on

Fail-friendly environment

Confession failures

Hack Time
 $\approx 10\%$

People are natural innovators

Spotify Hack Week

Do whatever! with whoever!
In whatever way!

Make cool things real!
Demos & Party on Fridays!

Minimize the need for Big Projects

Big Project = Big Risk

Chaos > Bureaucracy

Agile

Bureaucracy

Weekly Demo

Visual Progress

Keep

- Retrometives
- Daily Standup
- Continuous Delivery
- Git
- Git Branches

Skip/Jump

- Time reports
- Handovers
- Separate test teams
- Test phases
- Tool estimates
- Unless meetings
- Corporate BS

Chaos

Culture

Getting Started

YouTube Play Button

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.

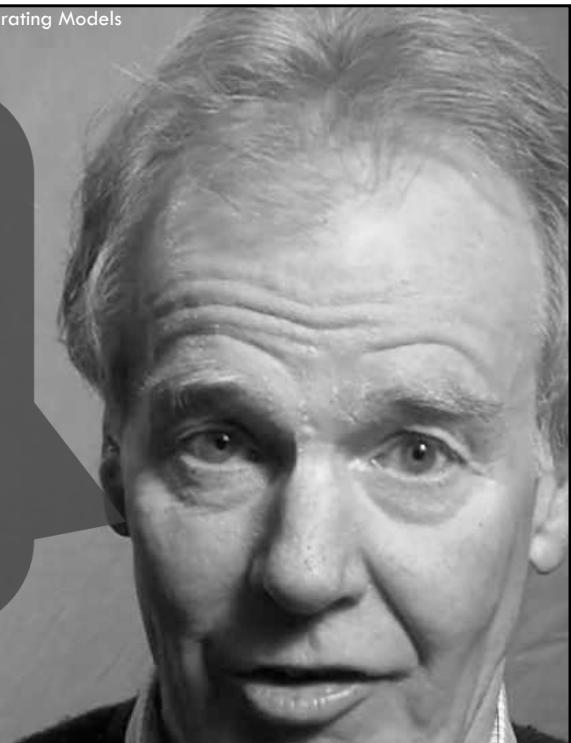
128

EXERCISE

Rating & Improving Using the Westrum Model

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

Peter Senge

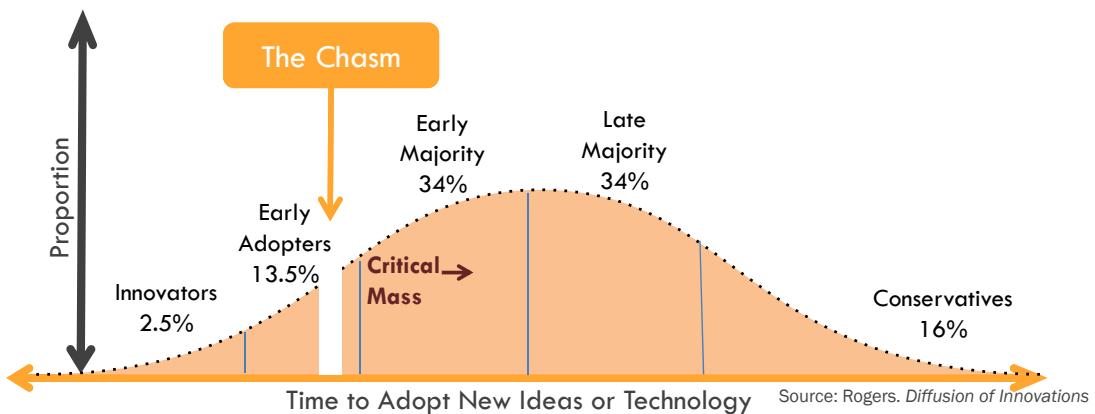


Culture Change is Never Easy

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



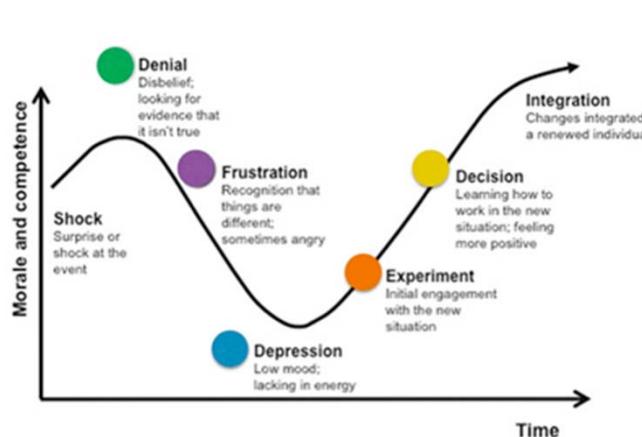
Adoption means that a person does something differently than before.

132

DISCUSSION

Placing on the change curve

The Stages of Change Acceptance



The Kübler-Ross Change Curve

From leanchange.org

134

© DevOps Institute unless otherwise stated

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?

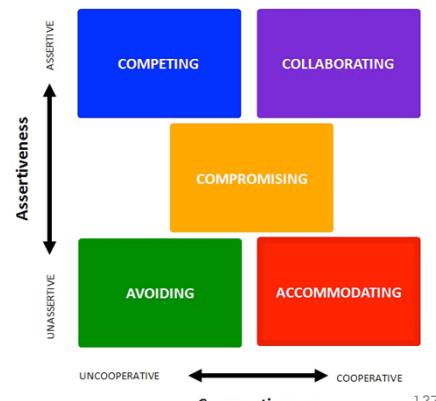
Expect Some Conflict: Thomas-Kilmann Conflict Modes

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

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 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.”



Heather Mickman,
Transformative
Technology Executive

Ross Clanton, Head of
Engineering

“We are a
technology
company.”

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

© DevOps Institute unless otherwise stated

Module 5: Quiz Answers

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

© DevOps Institute unless otherwise stated

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 Institute unless otherwise stated

144



“DevOps is not about automation, just as astronomy is not about telescopes.”

Christopher Little, quoted in The DevOps Handbook

Module 6: Automation & Architecting Toolchains

PERIODIC TABLE OF DEVOPS TOOLS (v3)

The Periodic Table of DevOps Tools is a grid of 180 tools, each represented by a colored square with its name and a brief description. The tools are categorized into groups based on their primary function and characteristics. A legend at the top right provides a key for these categories.

Category	Tool	Description
Source Control Mgmt.	GitLab (Gl)	GitLab
Database Automation	Databricks (Db)	Databricks
Continuous Integration	Jenkins (Jn)	Jenkins
Testing	JUnit (Ju)	JUnit
Enterprise	DBMaestro (Db)	DBMaestro
Open Source	Codewise (Cs)	Codewise
Free	FitNesse (Fn)	FitNesse
Freemium	Karma (Ka)	Karma
Paid	SoapUI (Su)	SoapUI
Configuration	Chef (Ch)	Chef
Monitoring	Terraform (Tf)	Terraform
Deployment	XebiaLabs Xl Deploy (Xld)	XebiaLabs Xl Deploy
Containers	Docker (Dk)	Docker
Release Orchestration	UrbanCode Release (Ur)	UrbanCode Release
Cloud	Azure Functions (Af)	Azure Functions
Analytics	Kubernetes (Ku)	Kubernetes
Security	Lambda (Ld)	Lambda
Collaboration	Sumo Logic (Sg)	Sumo Logic
AIOps	IBM Cloud (Fd)	IBM Cloud
En	Splunk (Sp)	Splunk
Fm	Alibaba Cloud (Al)	Alibaba Cloud
Pd	OpenStack (Os)	OpenStack
En	Prometheus (Ps)	Prometheus
Fm	Cloud Foundry (Cy)	Cloud Foundry
Pd	ITRS (It)	ITRS
En	Apache OpenWhisk (Aw)	Apache OpenWhisk
Fm	Logstash (Ls)	Logstash
Os	HashiCorp Vault (Hv)	HashiCorp Vault
Os	XebiaLabs Enterprise DevOps (XL)	XebiaLabs Enterprise DevOps
Os	ServiceNow (Sw)	ServiceNow
Os	Jira (Jr)	Jira
Os	Trello (Ti)	Trello
Os	Slack (Sk)	Slack
Os	Stride (St)	Stride
Os	CollabNet VersionOne (Cn)	CollabNet VersionOne
Os	Remedy (Ry)	Remedy
Os	Agile Central (Ac)	Agile Central
Os	OpsGenie (Og)	OpsGenie
Os	PagerDuty (Pd)	PagerDuty
Os	Smart (Sn)	Smart
Os	Triptire (Tw)	Triptire
Os	CyberArk (Ck)	CyberArk
Os	Veracode (Vc)	Veracode
Os	Fortify SCA (Ff)	Fortify SCA

Follow @xebialabs

https://xebialabs.com/periodic-table-of-devops-tools

Correlations Between Automation & High Performance

High performers have:

- 46 times more frequent code deployments
- 440 times faster lead time from commit to deploy
- 96 times faster mean time to recover from downtime
- 5 times lower change failure rate (changes are 1/5 as likely to fail)

From The State of
DevOps Report 2017

IT performance metrics rate	2016	2017
Deployment frequency	200x more frequent	46x more frequent
Lead time for changes	2,555x faster	440x faster
Mean time to recover	24x faster	96x faster
(MTTR) Change failure	3x lower (1/3 as likely)	5x lower (1/5 as likely)

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

The image shows a screenshot of a YouTube video player. At the top, there's a diagram illustrating the DevOps Toolchain. It consists of four main components arranged in a square: "SOURCE CONTROL / VERSIONING" (with a person icon), "CONTINUOUS INTEGRATION" (with a circular arrow icon), "DEVELOPMENT ENVIRONMENT" (with a wrench and screwdriver icon), and "CONFIGURATION MANAGEMENT" (with two interlocking gears). Arrows connect them in a clockwise cycle, labeled "INTEGRATION". Below the diagram, the text "ES" is followed by an arrow pointing right, and "GIT, SVN, TFS, CVS," is listed. To the right of the diagram, "DRONE.IO, JENKINS, TEAM CITY, TRAVIS CI" is listed. In the top right corner of the video player, there's a "Up next" section showing three more video thumbnails:

- How Netflix Does DevOps** (29:33) - A thumbnail for a Netflix video.
- Top 10 DevOps Tools** (7:35) - A thumbnail for a video by edureka!.
- What is DevOps? English** (7:07) - A thumbnail for a video by Rackspace.

Below the diagram, there's a large red play button icon. To its right, a dark grey rounded rectangle contains the title of the video: **The DevOps Toolchain with John Okoro (7:43)**. At the bottom of the video player, there's a progress bar showing the video has been watched for 5:38 minutes. Other controls like "SUBSCRIBED 715" and a notification bell are also visible.

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

Cloud computing: The practice of using remote servers hosted on the internet to host applications rather than local servers in a private datacenter.

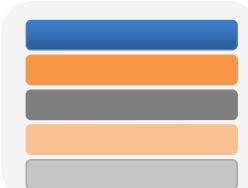


docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

© DevOps Institute unless otherwise stated



Teams that adopt essential cloud characteristics are **23 times** more likely to be elite performers.



Monolithic



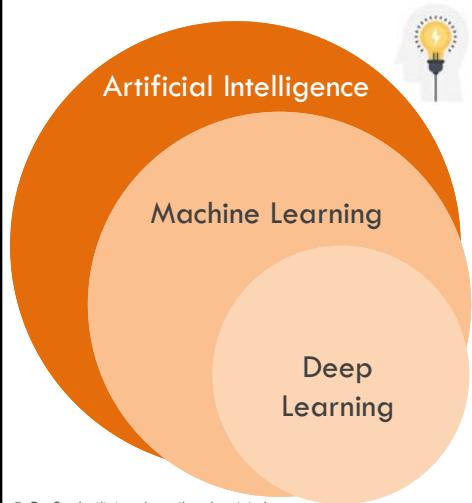
Containers & Microservices

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

151

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

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

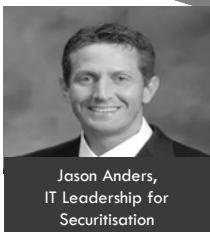
Do not underestimate the effort and cost of building toolchains from open source applications.
Open source is not necessarily free. It means that you can modify the source to fit your needs.

155

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

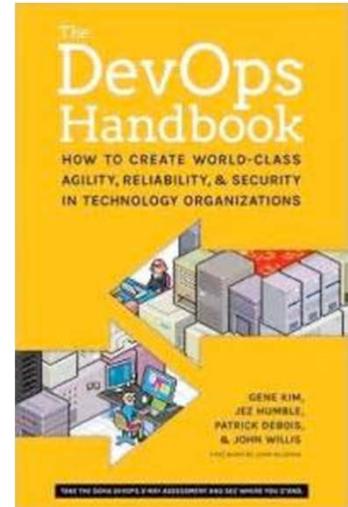
Benefits

- 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.”

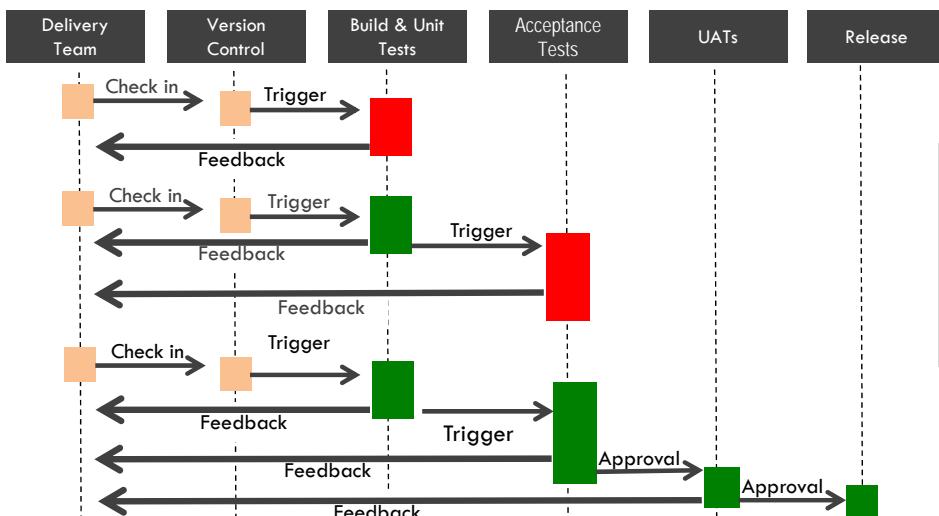
The DevOps Handbook



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

160

© DevOps Institute unless otherwise stated

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)

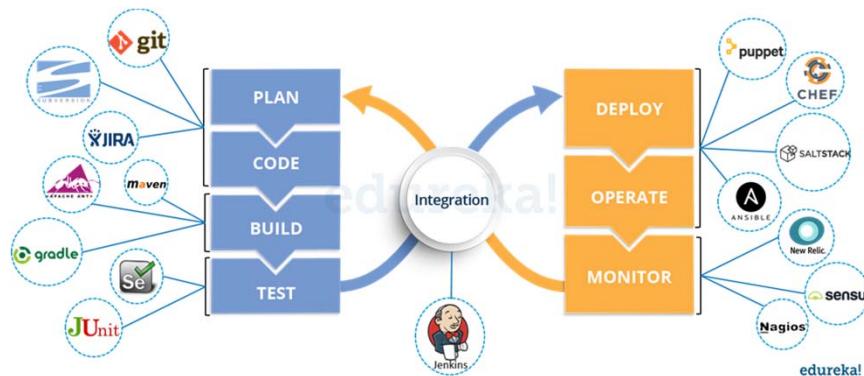
- Toolchains automate tasks in the deployment pipeline
- Each element of the toolchain serves a specific purpose
- Applications within the toolchains are connected via APIs
- They do not have to be homogenous or from a single vendor
- Toolchains are usually built around open and closed source ecosystems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

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

Automate Releases

Add more automation as needed

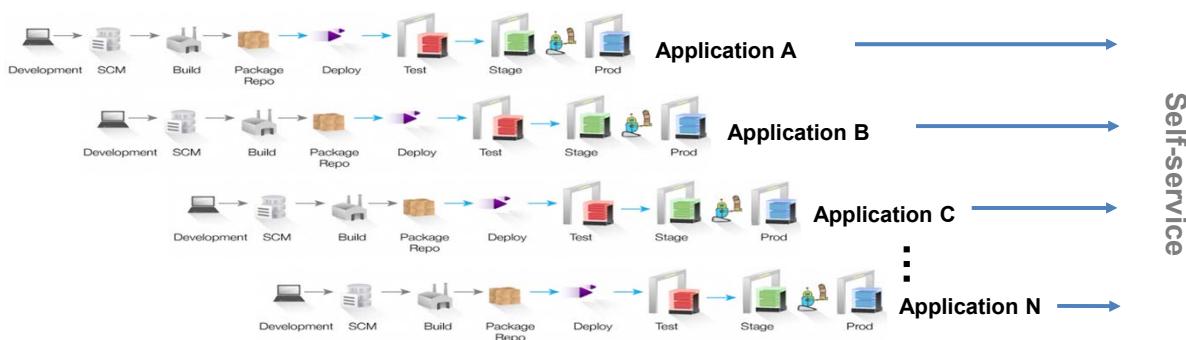
Ensure that each DevOps team member understands the capabilities and role of each tool in the DevOps toolchain to avoid tool overlap and toolchain functionality gaps.

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

© DevOps Institute unless otherwise stated

165

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 |

© DevOps Institute unless otherwise stated

Module 6: Quiz Answers

- | | | |
|---|--|---|
| 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 |

© DevOps Institute unless otherwise stated

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

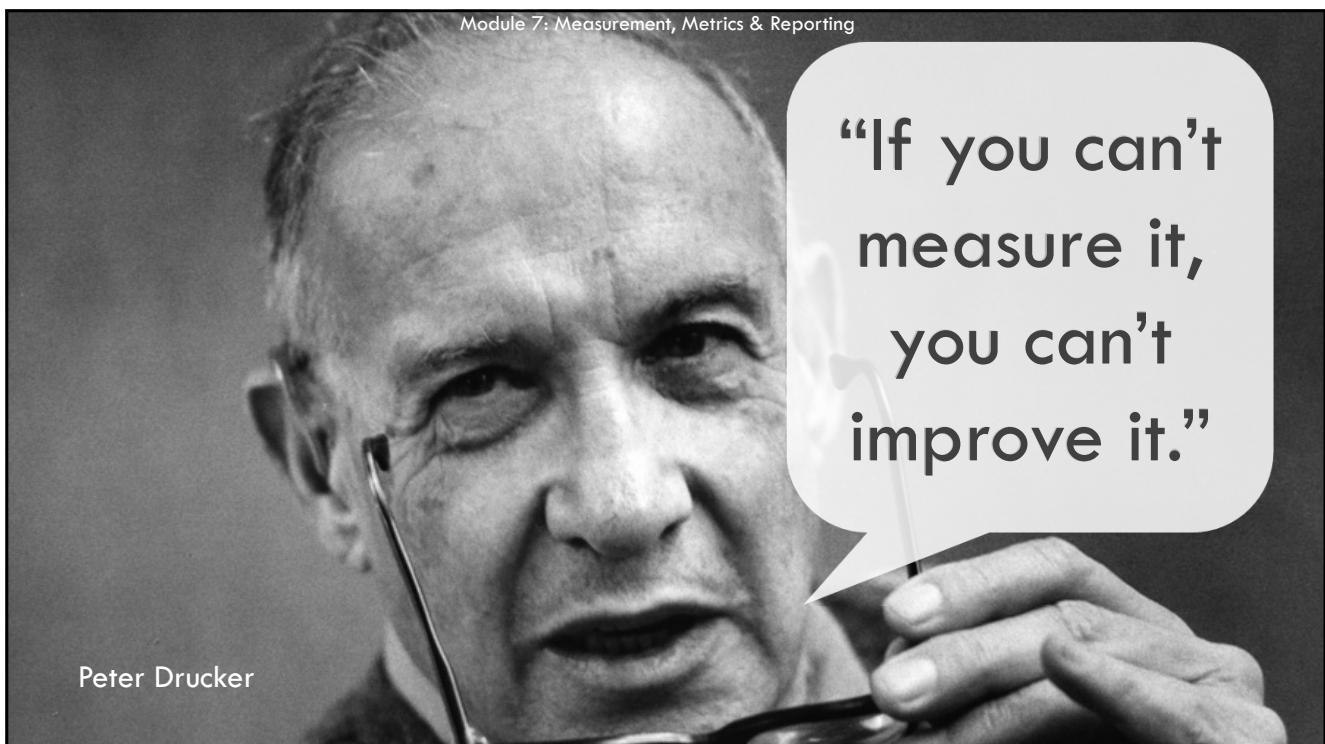
© DevOps Institute unless otherwise stated

The Importance of Measurement



The First Way	The Second Way	The Third Way
Flow	Feedback	Continuous Experimentation & Learning
<ul style="list-style-type: none"> • Change lead time • Change cycle time • Time to value • Value realisation 	<ul style="list-style-type: none"> • Build/test results • Change fail rate • Monitoring • % rework / complete & accurate 	<ul style="list-style-type: none"> • Hypothesis log • Time allocated • Time spent • Mastery achieved and reported
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

© DevOps Institute unless otherwise stated



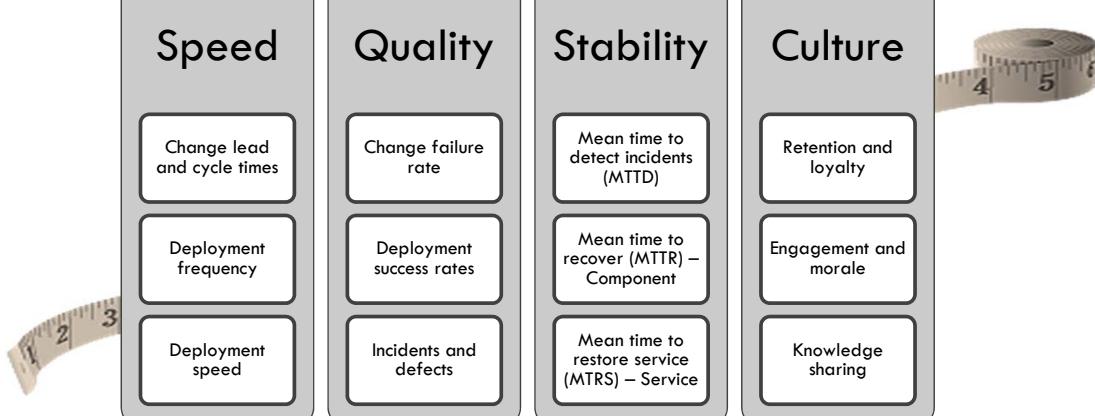
Peter Drucker

**“If you can’t
measure it,
you can’t
improve it.”**



Licensed For Use Only By: Serge PETIT sergeplace@gmail.com Sep 11 2019 11:42AM

Measuring Success



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

Adapted from
Splunk 2016

174

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

Accelerate: Dr Nicole Forsgren, Jez Humble & Gene Kim

4 types of (IT) work:

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

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?”



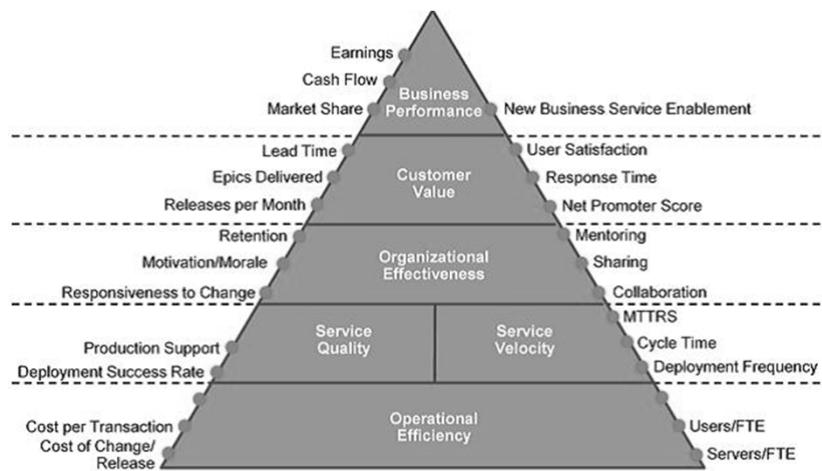
Carlos Gonsalves, Global
Chief Technology

“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



© DevOps Institute unless otherwise stated

EXERCISE

The Most Meaningful Metrics

Module 7: Quiz

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

© DevOps Institute unless otherwise stated

Module 7: Quiz Answers

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

© DevOps Institute unless otherwise stated

MODULE 8

Sharing, Shadowing and 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 Institute unless otherwise stated

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.

185

Module 8: Sharing, Shadowing & Evolving

youtube.com

The video player displays a video thumbnail of a man with glasses speaking in an office environment. The title 'DevOps: A Culture of Sharing' is overlaid on the video frame. Below the video, there is a large red play button icon.

Up next

- DEVOPS** What is DevOps? CBT Nuggets 98K views 9:01
- The Culture Change of John Okoro 2.8K views 6:51
- What is DevOps? - In Simple English Rackspace 891K views 7:07
- CHARLIE MUNGER Charlie Munger Interview 2018

PUPPET Published on Apr 25, 2016

SUBSCRIBE 12K

49:49 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?

Module 8: Sharing, Shadowing & Evolving

DevOps in the Enterprise

CASE STORY: Disney

“There’s no secret to digital magic. We keep moving forward, opening 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)

Module 8: Sharing, Shadowing & Evolving

Roles

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

Todays CIOs are looking for workers who can shift gears and adapt to changing technology.

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

- DevOps evangelist or leader
- Software engineers, developers and testers
- Release manager
- 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 other roles do you think should be involved?

Source: <http://techbeacon.com/7-devops-roles-you-need-succeed>

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



Module 8: Sharing, Shadowing & Evolving

DevOps Leadership

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.

From The State of DevOps Report 2017

197

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

Crucial Ingredients	Leadership Challenges
<ol style="list-style-type: none">1. Collaboration - break down silos, mutual objectives2. Curiosity - keep experimenting3. Courage - candor, challenge, no blaming or witch-hunting	<ul style="list-style-type: none">• The politics of command and control• How new leadership can take a company in a new direction• The blame bias of who versus what

Module 8: Sharing, Shadowing & Evolving

Organizational Considerations

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.

200

DevOps Teams (1)

The creation of DevOps departments or teams was a growing trend; 16% in 2014, 19% in 2015, and 22% in 2016, and 27% of 2017 AND 2018 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

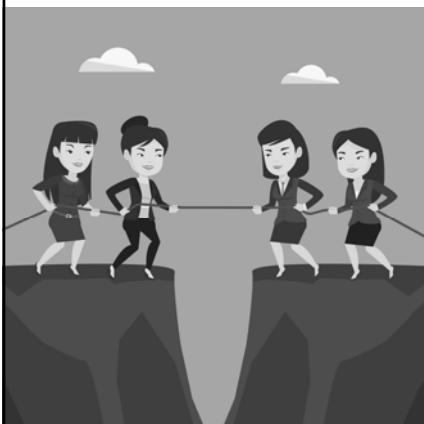
There is no ‘ideal’ structure for a DevOps team.

201

© DevOps Institute unless otherwise stated



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.

202

Module 8: Sharing, Shadowing & Evolving

Getting Started

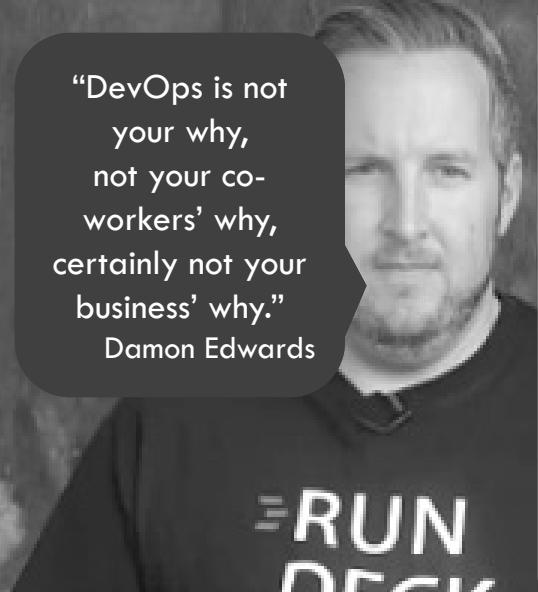
“It’s a journey, not a silver bullet, and leaders need to avoid getting caught in analysis paralysis. Start making the changes, get the wins and let the organization evolve.”

Melissa Sargeant



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

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

206

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

207

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

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

210

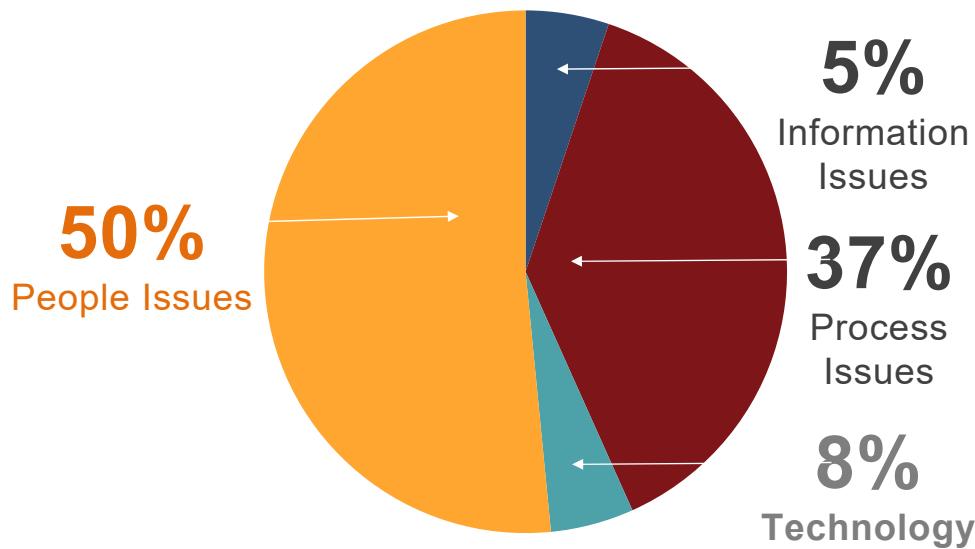
Challenges and Risks

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?



© DevOps Institute unless otherwise stated

Source: Gartner

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

© DevOps Institute unless otherwise stated

Module 8: Quiz Answers

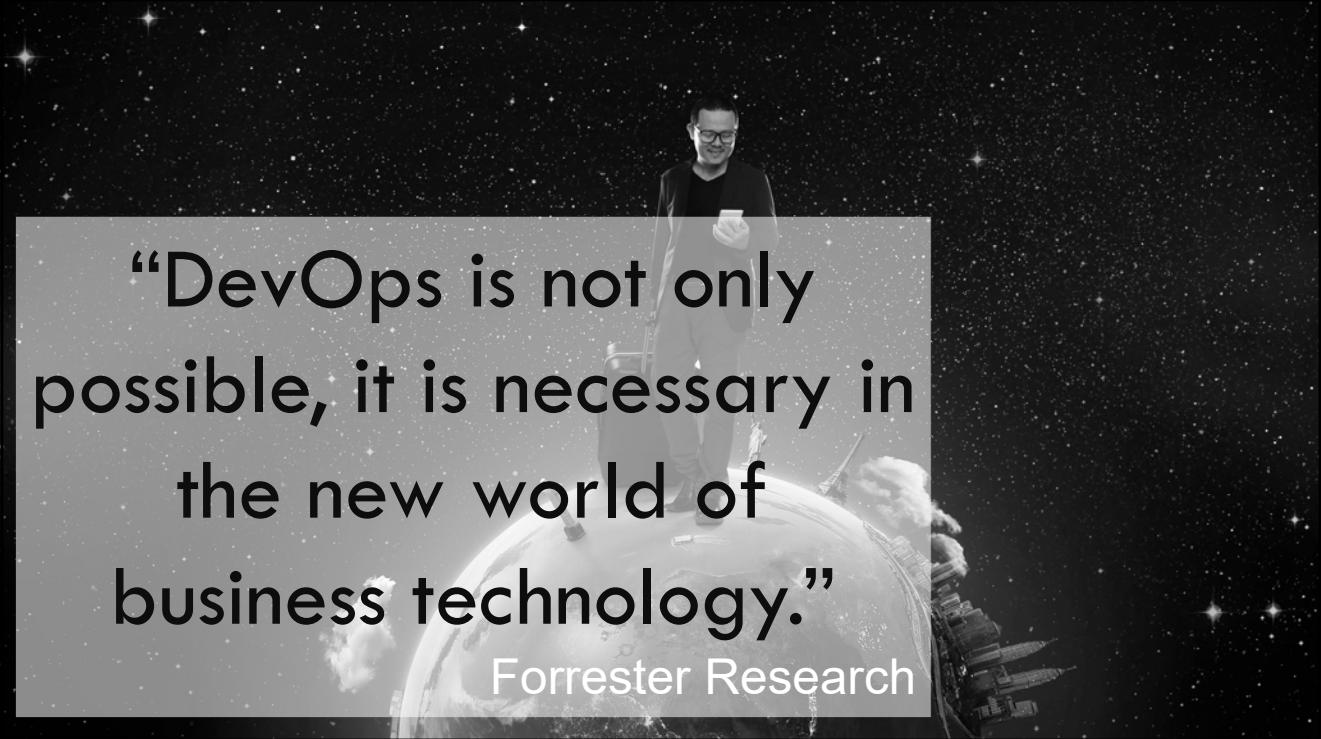
- | | |
|---|--|
| 1 Who provides 'lofts' as an immersive experience? | a) Amazon
b) IBM
c) Google
d) Puppet |
| 2 What's an 'Ignite' in the context of DevOps Days? | a) A key note
b) A 30 minute presentation
c) A 5 minute topic specific session
d) A breakout session |
| 3 Why do Disney keep doing more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

© DevOps Institute unless otherwise stated

Summary

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

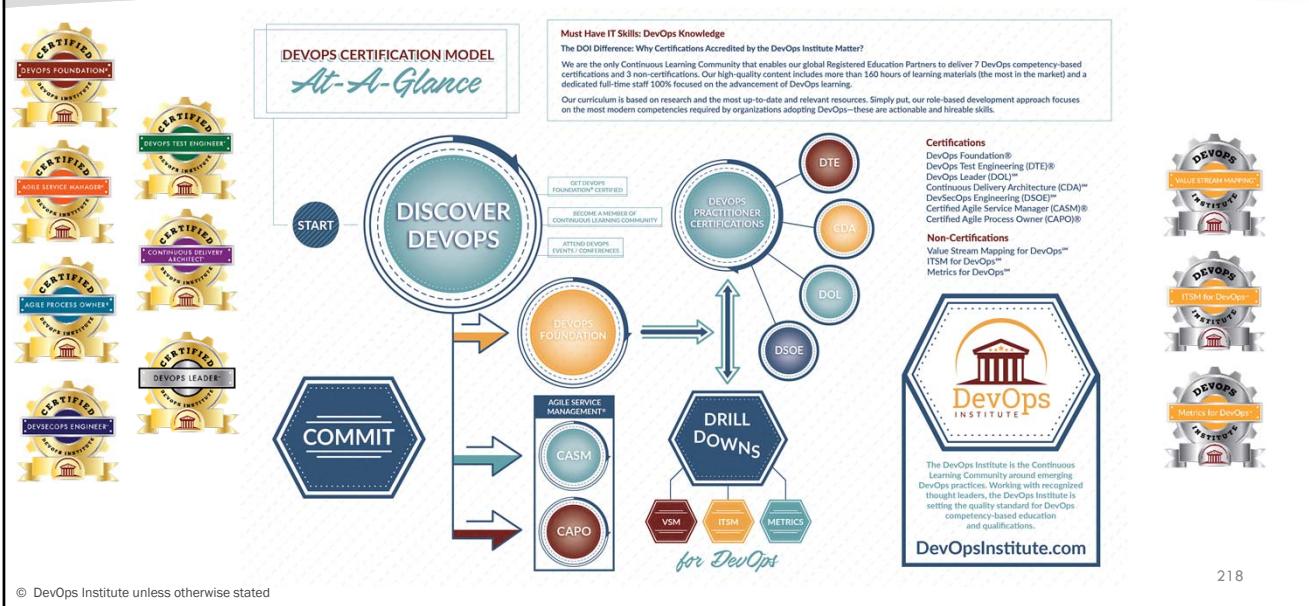
- Improving communication, collaboration and the integration of processes and tools across the IT value stream
- Automating the process of software delivery and infrastructure changes
- Leveraging agile, lean, ITSM and evolving DevOps practices



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

Forrester Research

DOI Learning Tracks & Community





DevOps Foundation®

Examination Requirements



DevOps Foundation Certificate

DevOps Foundation is a freestanding certification from the DevOps Institute. The purpose of this course and its associated exam is to impart, test and validate knowledge of DevOps basic vocabulary, principles and practices. The vocabulary terms, concepts and practices are documented in the course learner manual. DevOps Foundation is intended to provide individuals an understanding of basic DevOps concepts and how DevOps may be used to improve communication, collaboration and integration between software developers and IT operations professionals.

Eligibility for Examination

Although there are no formal prerequisites for the exam, DevOps Institute highly recommends the following to prepare candidates for the exam leading to DevOps Foundation certification:

- It is recommended that candidates complete at least 16 contact hours (instruction and labs) as part of a formal, approved training course delivered by an approved Registered Education Partner of the DevOps Institute
- It is recommended that students complete at a minimum 6 hours of personal study by reviewing the vocabulary list and pertinent areas of the course learner manual and by completing the sample exam

Examination Administration

The DevOps Foundation examination is accredited, managed and administered under the strict protocols and standards of the DevOps Institute.

Level of Difficulty

The DevOps Foundation certification uses the Bloom Taxonomy of Educational Objectives in the construction of both the content and the examination.

- The DevOps Foundation exam contains Bloom 1 questions that test learners' **knowledge** of DevOps concepts and vocabulary terms (see list below)
- The exam also contains Bloom 2 questions that test learner's **comprehension** of these concepts in context

Format of the Examination

Candidates must achieve a passing score to gain the DevOps Foundation Certificate.

Exam Type	40 multiple choice questions
Duration	60 minutes for candidates in their respective language 75 minutes if English is not a candidate's native tongue
Prerequisites	DevOps Foundation course from a DevOps Institute Registered Education Partner (REP)
Supervised	Yes
Open Book	No
Passing Score	65%
Delivery	Online or invigilated in person
Badge	DevOps Foundation Certified

Exam Topic Areas and Question Weighting

The DevOps Foundation exam requires knowledge of the topic areas described below.

Module	Description	Max Questions
DOFD – 1 Exploring DevOps	Purpose, objectives and business value of DevOps	5
DOFD – 2 Core DevOps Principles	The Three Ways	4
DOFD – 3 Key DevOps Practices	Emerging DevOps Practices such as continuous delivery and continuous integration	7
DOFD – 4 Business & Technology Frameworks	The relationship between relevant frameworks and standards and DevOps	7
DOFD – 5 DevOps Values - Culture, Behaviors & Operating Models	Characteristics of a DevOps culture and of culture change	6
DOFD – 6 DevOps Values - Automation & Architecting DevOps Toolchains	The Deployment Pipeline, DevOps toolchains and other automation considerations	5
DOFD – 7 DevOps Values – Measurement, Metrics & Reporting	Common DevOps practices and related processes	2
DOFD – 8 DevOps Values: Sharing, Shadowing and Reporting	Responsibilities of key roles and considerations relative to organizational structure. Getting started - adoption challenges, risks, critical success factors and key performance measures	4

Concept and Terminology List

After studying this course, the candidate is expected to understand the following DevOps concepts and vocabulary at a Blooms Level 1 and 2.

- Agile Manifesto
- Agile service management
- Agile software development
- Application Programming Interface (API)
- CALMS
- Change failure rate
- Change fatigue
- Change lead time
- ChatOps
- Code commit
- Collaboration and communication
- Collective Body of Knowledge (CBOK)
- Configuration management tools
- Conflict management
- Constraint
- Containers
- Continuous integration
- Continuous delivery
- Continuous deployment
- Cultural debt
- Cycle time
- Deployment pipeline
- Deployment frequency
- DevSecOps
- DevOps
- DevOps metrics
- DevOps stakeholders
- DevOps teams
- DevOps roles
- Flow
- Golden Circle
- High-performing organizations
- Impediment
- Immersion
- Improvement kata
- IT service management
- Kanban
- Lean production
- Lean thinking
- Lean tools
- Lean types of Waste (DOWNTIME)
- Microservices
- Open source
- Organizational culture
- Organizational considerations
- Outcome economy
- Pace-Layered Application Strategy
- Process Model
- Rugged DevOps
- Scaled Agile Framework (SAFe)
- Scrum
- Scrum roles, artifacts and events
- Sharing opportunities
- Service
- Shift left
- Simian Army/Chaos Monkey
- Test driven development
- Testing (unit, acceptance, integration)
- The Three Ways
- Theory of Constraints
- DevOps toolchain
- Value stream mapping
- Variable speed IT
- Velocity
- Waste
- Waterfall



DevOps Foundation V3.10

Sample Examination

1. A small group of individuals recently returned from a conference where they learned about DevOps. They cannot agree on how to get started. Where should an IT organization start when adopting DevOps practices?

- A. Understand why the organization exists
- B. Pick the right applications to pilot
- C. Develop a long-term strategy
- D. Identify tools and training needed

2. What is the Three Ways?

- A. Methodology for identifying and removing constraints
- B. The key principles of DevOps
- C. Disciplined, data-driven approach for reducing waste
- D. A methodology for performing continuous improvement

3. Which statement about Kanban is CORRECT?

- A. Pushes work through a process
- B. Requires a workflow management tool
- C. Pulls work through a process
- D. Enables more work in progress

4. What is the Agile Manifesto?

- A. Values and principles to guide an iterative and people-centric approach to software development
- B. Methodology that focuses on making sure software is always in a releasable state throughout its lifecycle
- C. Declaration of the benefits and intentions of DevOps
- D. Intentions and motives of being an agile enterprise

5. An organization is trying to overcome the challenges of their legacy silo culture where teams have been organized by subject matter expertise. What is this organization suffering from?

- A. Cultural debt
- B. Change fatigue
- C. Organizational change
- D. Low trust

6. Which statement **BEST describes change fatigue?**

- A. Aggressive resistance
- B. Apathy
- C. Finger pointing
- D. Exhaustion

7. Due to a tightly-coupled architecture, an organization is unable to increase the frequency of releases for a key service. When releases do occur, they are extremely painful and the organization's competitive advantage is eroding as a result. Which software development approach could be used to improve this situation?

- A. Test-driven development
- B. Containers
- C. Microservices
- D. Chaos Monkey

8. An organization has just completed the deployment of a pilot release using DevOps practices and a preliminary deployment pipeline. Which metric would provide the most information to help them continually improve?

- A. Mean Time to Repair (MTTR)
- B. Change lead and cycle times
- C. Knowledge sharing
- D. All of the above

9. Which statement about DevOps teams is **MOST accurate?**

- A. They are responsible for establishing DevOps practices across the enterprise
- B. They are accountable for the development of the deployment pipeline
- C. They should be a fixed team that works together on long term projects
- D. They should have shared accountabilities

10. An organization recently held an internal DevOps Days. During one of the openspace sessions, it was suggested that there be more opportunities for dev, ops, security and other IT areas to interact and share. What sort of opportunities should the organization consider?

- A. Hackathons
- B. Simulations
- C. Immersion opportunities
- D. All of the above

11. Which of the following roles are DevOps stakeholders?

- A. QA testers
- B. Support professionals
- C. Suppliers
- D. All of the above

12. Which is NOT a goal of DevOps?

- A. Improved productivity
- B. Fewer but higher-quality software releases
- C. Lower risk software deployments
- D. Improved quality of code

13. An organization is implementing a disruptive application similar to the Simian Army. Which of the Three Ways are they introducing?

- A. The First Way
- B. The Second Way
- C. The Third Way
- D. The Phoenix Project

14. An organization is looking to improve real-time collaboration between teams. Which DevOps practice should they be considering?

- A. Kanban
- B. ChatOps
- C. Escalation
- D. Alerts

15. Which is a characteristic of a DevOps culture?

- A. Effective one-way communication from the top down
- B. Recognizing the best and brightest for their successes
- C. Shared vision, goals and incentives
- D. All of the above

16. Which BEST describes a deployment pipeline?

- A. Automation version of the ITSM change management process
- B. Automated process for managing software changes from check-in to release
- C. Collection of tools that enable continuous integration
- D. Sequence of value-adding activities required to design, build and deliver a product

17. At a recent conference, a CIO was told that her organization should invest heavily in machine learning. Back at the office, she asked one of her senior leadership team to pull together an investment case. What is unlikely to be a direct benefit they are likely to receive from using artificial intelligence and should not be included in the investment case?

- A. Predicting future scenarios
- B. Finding new trends and correlations
- C. Augmenting human contribution and boosting productivity
- D. Building a blame free culture

18. Which statement about the Improvement Kata is CORRECT?

- A. It focuses on short term goal
- B. It is a 7-step process
- C. It considers the organization's long-term vision or direction
- D. It should be performed as time allows

19. In the context of agile software development, which of the following is NOT a responsibility of IT Operations?

- A. Managing the product backlog
- B. Defining non-functional requirements
- C. Identifying security requirements
- D. Provisioning the infrastructure

20. Which of the following is a characteristic of a high trust organizational culture?

- A. Good information flow
- B. Cross-functional collaboration
- C. Learning from failures and new ideas
- D. All of the above

21. Why is organizational culture a critical success factor for DevOps?

- A. It represents the values and behaviors that contribute to the unique social and psychological environment of an organization
- B. It represents a command and control approach to the delivery of services
- C. It represents the way that an organization is structured and organized
- D. It reflects the strategic direction of the business' leadership

22. What is a primary benefit of DevOps toolchains?

- A. To automate steps in the deployment pipeline
- B. To trace features' journeys from inception to live
- C. To ensure that the architectural design supports interoperability and consistency
- D. All of the above

23. Which of the following DevOps roles is NOT YET well defined?

- A. Systems engineer
- B. Continuous delivery automation architect
- C. DevOps engineer
- D. Experience assurance

24. An organization is implementing DevOps. The developers concerned that their ITSM processes are too complex, slow and will not support DevOps principles and practices. Which IT framework will help the organization instill agile thinking into existing ITSM processes?

- A. ITIL
- B. Agile
- C. Agile service management
- D. Lean

25. Which lean tool depicts the flow of information, materials and work across functional silos with an emphasis on quantifying and eliminating waste?

- A. Improvement Kata
- B. Continuous Delivery
- C. Kanban
- D. Value stream mapping

26. The business ‘why’ of the Golden Circle represents an organization’s

- A. Purpose, cause and belief
- B. Products and services
- C. Competitive advantage
- D. Profitability

27. What determines which DevOps principles and practices will BEST benefit an organization?

- A. Business strategies and goals
- B. The commitment of early adopters
- C. The availability of advanced tools
- D. IT's capabilities and resources

28. The Theory of Constraints supports which of the Three Ways?

- A. The First Way
- B. The Second Way
- C. The Third Way
- D. All of the above

29. Which of the following is required for Continuous Integration?

- A. Automated unit, integration and acceptance testing
- B. Automated release management
- C. Continuous delivery pipeline
- D. Deployment pipeline

30. Which DevOps practice relies on a deployment pipeline that enables push-button deployments on demand?

- A. Continuous testing
- B. Continuous integration
- C. DevSecOps
- D. Continuous delivery

31. Which of the following ITSM processes are most critical to DevOps?

- A. Change, Release and Service Asset and Configuration Management
- B. Incident, Problem and Event Management
- C. Demand, Capacity and Information Security Management
- D. All of the above

32. An organization has identified they have a culture of blame and fear, where incidents are not valued and failure is not embraced as a learning opportunity. There are many single points of failure and employees suffer daily as a result of the fragility of the systems, enduring painful war-rooms during frequent outages. What should this organization look to in order to improve the situation?

- A. Safety Culture
- B. Agile software development
- C. Building a DevOps toolchain
- D. Site Reliability Engineering

33. When trying to effect major change, who should be engaged in planning activities and serve as change agents?

- A. Early adopters
- B. Naysayers
- C. Management
- D. People who need proof

34. What of the following is NOT a typical element in a DevOps toolchain?

- A. Monitoring tools
- B. Test automation
- C. Version control
- D. Service desk incident management systems

35. Which of the following is a critical success factor for DevOps?

- A. Establishing a tool chain
- B. Hiring DevOps Engineers
- C. Management commitment to culture change
- D. Automating everything

36. Which of the following is not a goal of DevOps leadership?

- A. Help to improve self-diagnosis
- B. Control and evaluate workers using metrics
- C. Instil self-improvement
- D. Translate local discoveries to global improvements

37. Which of the Three Ways encourages peer review of production changes?

- A. The First Way
- B. The Second Way
- C. The Third Way
- D. All of the above

38. What does the concept of “shift left” represent?

- A. Building quality into the software development process via early and continuous testing
- B. Passing release packages to IT Operations following completion of a batch of development
- C. Performing random tests on code that is committed to a continuous integration server
- D. Doing more testing in production after deployment

39. An organization is struggling with the additional time it takes for security reviews after an Agile team completes a Sprint. The delay is impacting their ability to release. They want to include more security testing as part of their “shift left” testing approach. Which DevOps practice would they need?

- A. ChatOps
- B. Continuous Testing
- C. DevSecOps
- D. Vulnerability alerts

40. An organization is preparing to automatically deploy every release that passes automated unit, integration, user acceptance and non-functional tests. Which DevOps practice are they applying?

- A. Continuous delivery
- B. Continuous testing
- C. Continuous deployment
- D. Continuous integration

ANSWER KEY

Question	Correct Answer	Topic Area
1	A	1: Exploring DevOps
2	B	2: Core DevOps Principles
3	C	3: Key DevOps Practices
4	A	4: DevOps Values: Business & Technology Frameworks
5	A	5: DevOps Values: Culture, Behaviors & Operating Models
6	B	5: DevOps Values: Culture, Behaviors & Operating Models
7	C	6: DevOps Values: Automation & Architecting Toolchains
8	D	7: DevOps Values: Measurements, Metrics & Reporting
9	D	8: DevOps Values: Sharing, Shadowing & Evolving
10	D	7: DevOps Values: Measurements, Metrics & Reporting
11	D	1: Exploring DevOps
12	B	1: Exploring DevOps
13	C	2: Core DevOps Principles
14	B	3: Key DevOps Practices
15	C	5: DevOps Values: Culture, Behaviors & Operating Models
16	B	6: DevOps Values: Automation & Architecting Toolchains
17	D	6: DevOps Values: Automation & Architecting Toolchains
18	C	4: DevOps Values: Business & Technology Frameworks
19	A	4: DevOps Values: Business & Technology Frameworks
20	D	5: DevOps Values: Culture, Behaviors & Operating Models
21	A	5: DevOps Values: Culture, Behaviors & Operating Models
22	D	6: DevOps Values: Automation & Architecting Toolchains
23	C	8: DevOps Values: Sharing, Shadowing & Evolving
24	C	4: DevOps Values: Business & Technology Frameworks

Question	Correct Answer	Topic Area
25	D	4: DevOps Values: Business & Technology Frameworks
26	A	1: Exploring DevOps
27	A	1: Exploring DevOps
28	A	2: Core DevOps Principles
29	A	3: Key DevOps Practices
30	D	3: Key DevOps Practices
31	D	4: DevOps Values: Business & Technology Frameworks
32	A	4: DevOps Values: Business & Technology Frameworks
33	A	5: DevOps Values: Culture, Behaviors & Operating Models
34	D	6: DevOps Values: Automation & Architecting Toolchains
35	C	8: DevOps Values: Sharing, Shadowing & Evolving
36	B	8: DevOps Values: Sharing, Shadowing & Evolving
37	B	2: Core DevOps Principles
38	A	3: Key DevOps Practices
39	C	3: Key DevOps Practices
40	C	3: Key DevOps Practices



DevOps Foundation

Value Add Resources and Source Materials

DevOps Foundation Course: Value Added Resources

This document provides links to articles and videos related to the DevOps Foundation course from the DevOps Institute. This information is provided to enhance your understanding of DevOps Foundation-related concepts and terms and is not examinable. Of course, there is a wealth of other videos, blogs and case studies on the web. We welcome suggestions for additions.

Videos Featured in the Course

Module Featured	Title & Description	Link
1: Exploring DevOps	'A Short History of DevOps' with Damon Edwards (11:47)	https://youtu.be/o7-luYS0iSE
1: Exploring DevOps	Abbreviated version of Simon Sinek's Ted Talk 'Start with Why - How Great Leaders Inspire Action' (5:00)	https://youtu.be/lPYeCltXpxw
2: Core DevOps Practices	'Gene Kim Defines The Three Ways of The Phoenix Project' (3:31)	https://youtu.be/nUOXDEvpIRc
3: Key DevOps Principles	'GitHub Professional Guide: Continuous Integration & Continuous Delivery' (6:00)	https://youtu.be/xSv_m3KhUO8
4: Business & Technology Frameworks	'Spotify Engineering Culture Part 1' with Henrik Kniberg (13:12)	https://youtu.be/4GK1NDTWbkY
5: DevOps Values: Culture, Behaviors & Operating Models	'Spotify Engineering Culture Part 2' with Henrik Kniberg (13:27)	https://youtu.be/rzoiryY2STQ
6: DevOps Values: Automation & Architecting DevOps Toolchains	'The DevOps Toolchain' with John Okoro (7:43)	https://youtu.be/bwE8aFPAzj8
7: DevOps Values: Measurement, Metrics & Reporting	'4 DevOps Metrics to Improve Delivery Performance' with Dennis Ehle (6:05)	https://youtu.be/RwvgEz_4IYI
8: DevOps Values: Sharing, Shadowing & Evolving	'DevOps: A Culture of Sharing' with Gareth Rushgrove (2:19)	https://youtu.be/8aJHtjp-3U

DevOps Reports

Report Name	Writers/Publishers	Link
2018 DevSecOps Community Survey	Sonatype	https://www.sonatype.com/2018survey

DevOps Foundation Course: Value Added Resources

2018 Global Developer Report	Gitlab	https://about.gitlab.com/developer-survey/2018/
State of DevOps Report	InterOp	http://reg.interop.com/stateofdevops
State of the Cloud Report	RightScale	https://www.rightscale.com/lp/state-of-the-cloud
The Accelerate State of DevOps Report	Dr. Nicole Forsgren, Gene Kim & Jez Humble in collaboration with Google Cloud Platform (GCP)	https://cloudplatformonline.com/2018-state-of-devops.html
The State of DevOps Report	Dr. Nicole Forsgren, Gene Kim & Jez Humble in collaboration with Puppet Labs	https://puppet.com/resources/white-paper
The Value Unleashed by DevOps	Boston Consulting Group	https://www.bcg.com/en-us/agile/devops/benefits.aspx

DevOps Articles

Article Title & Author	Relevant Module	Link
'2018: The Year of Enterprise DevOps' by Forrester	1: Exploring DevOps	https://go.forrester.com/blogs/2018-the-year-of-enterprise-devops/
'5 Things DevOps is Not' from devops.com	1: Exploring DevOps	https://devops.com/what-devops-is-not/
'6 DevOps Recruiting Tips: How to Land the Right People' on TechBeacon	8: Sharing, Shadowing & Evolving	https://techbeacon.com/6-devops-recruiting-tips-how-land-right-people
'7 DevOps Roles You Need to Succeed' on TechBeacon	5: DevOps Values: Culture, Behaviors & Operating Models	https://techbeacon.com/7-devops-roles-you-need-succeed
'7 Keys to Finding Phenomenal DevOps Talent' on TechBeacon	8: Sharing, Shadowing & Evolving	https://techbeacon.com/7-keys-finding-phenomenal-devops-talent?es_p=1623370
'10 Ways Machine Learning Can Optimize DevOps' on TechBeacon	6: Automation & Architecting Toolchains	https://techbeacon.com/10-ways-machine-learning-can-optimize-devops
'A different drumbeat: using Kanban for devops to smooth out your scrum cycles' by Nate Berent-Spillson	3: Key DevOps Principles	https://www.infoworld.com/article/3266588/devops/a-different-drumbeat-using-kanban-for-devops-to-smooth-out-your-scrum-cycles.html

DevOps Foundation Course: Value Added Resources

'A Four Quadrant Look at the DevOps Toolchain' by Scott Johnston	6: Automation & Architecting Toolchains	https://puppet.com/blog/a-four-quadrant-look-at-devops-toolchain
'A Personal Reinterpretation of The Three Ways' by Tim Hunter	2: Core DevOps Practices	http://itrevolution.com/a-personal-reinterpretation-of-the-three-ways/
'A Typology of Organizational Cultures' by Ron Westrum	5: DevOps Values: Culture, Behaviors & Operating Models	http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf
Apply the Strangler Application Pattern to Microservices Applications - IBM	6: Automation & Architecting Toolchains	https://www.ibm.com/developerworks/cloud/library/cl-strangler-application-pattern-microservices-apps-trs/index.html
'Best Practices for Using Value Stream Mapping as a Continuous Improvement Tool' by R. Keith Mobley	4: Business & Technology Frameworks	http://www.industryweek.com/continuous-improvement/best-practices-using-value-stream-mapping-continuous-improvement-tool?page=2
Blue-green Deployments, A/B Testing, and Canary Releases by Christian Posta	3: Key DevOps Principles	http://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/
'Building a Healthy DevOps Culture' by Michael Butt	5: DevOps Values: Culture, Behaviors & Operating Models	https://www.wired.com/insights/2013/06/building-a-healthy-devops-culture/
'Building a Learning Organization' on HBR	4: Business & Technology Frameworks	https://hbr.org/1993/07/building-a-learning-organization
'ChatOps: Communicating at the Speed of DevOps' by George Hulme	6: Automation & Architecting Toolchains	http://devops.com/features/chatops-communicating-speed-devops/
'Codifying DevOps Practices' by Patrick DeBois	3: Key DevOps Principles	http://www.jedi.be/blog/2012/05/12/codifying-devops-area-practices
'Communities of Practice: The Missing Piece of Your Agile Organisation' by Emily Webber	5: DevOps Values: Culture, Behaviors & Operating Models	https://www.infoq.com/articles/communities-of-practice-agile-organisation
'Continuous Delivery' by Martin Fowler	3: Key DevOps Principles	http://martinfowler.com/bliki/ContinuousDelivery.html
'Continuous Delivery: Anatomy of the Deployment Pipeline' by Jez Humble & Dave Farley	3: Key DevOps Principles	http://www.informit.com/articles/article.aspx?p=1621865&seqNum=8
'Continuous Everything in DevOps...What is the difference	3: Key DevOps Principles	https://notafactoryanymore.com/2014/08/19/continuous-everything-in-devops-what-is-the-difference

DevOps Foundation Course: Value Added Resources

'between CI, CD, CD... ?' by Micro Hering		the-difference-between-ci-cdcd/
'Continuous Integration' on ThoughtWorks	3: Key DevOps Principles	https://www.thoughtworks.com/continuous-integration
'Cultural Debt Can Be the Primary Driver of Technical Debt' by Rick Brenner	5: DevOps Values: Culture, Behaviors & Operating Models	https://techdebtpolicy.com/cultural-debt/
'Culture Isn't a Swear Word' by Jonathan Fletcher	5: DevOps Values: Culture, Behaviors & Operating Models	https://devops.com/culture-isnt-swear-word/
'Creating a Higher Trust Organization' by John Mackey	5: DevOps Values: Culture, Behaviors & Operating Models	https://www.huffingtonpost.com/john-mackey/creating-the-high-trust-o_b_497589.html?guccounter=1
'Data-Driven DevOps: Use Metrics to Guide Your Journey' Gartner	7: Measurement, Metrics & Reporting	https://www.gartner.com/doc/3760663/data-driven-devops-use-metrics-guide
'DevOps and IT Support: 4 Principles to Keep Your Team Ahead of the Curve' on devops.com	1: Exploring DevOps	http://devops.com/2015/12/09/devops-and-it-support-4-principles-to-keep-your-team-ahead-of-the-curve/
'DevOps and Kanban - Match Made in Heaven' by UpGuard	3: Key DevOps Principles	https://www.upguard.com/blog/devops-kanban-match-heaven
'DevOps, Cloud and the Lean "Wheel of Waste"' by Richard Seroter	4: Business & Technology Frameworks	https://seroter.wordpress.com/2014/04/07/devops-cloud-and-the-lean-wheel-of-waste/
'DevOps Culture' by John Willis	5: DevOps Values: Culture, Behaviors & Operating Models	http://itrevolution.com/devops-culture-part-1/
'DevOps Demystified' by Ben Rockwood	7: Measurement, Metrics & Reporting	http://cuddletech.com/slides/DevOps-Demystified.pdf
'DevOps Requires New job Skills and Roles' by Upguard	8: Sharing, Shadowing & Evolving	https://www.upguard.com/blog/devops-new-job-skills-roles-titles
'DevOps: To Measure Value, Then To Measure Speed' by Stephen Williams	7: Measurement, Metrics & Reporting	https://www.linkedin.com/pulse/devops-measure-value-speed-stephen-williams/
'Doing ChatOps in Microsoft Teams' by Helen Beal	3: Key DevOps Principles	https://www.infoq.com/news/2018/02/chatops-microsoft-teams
'First Impressions at Etsy' by Jason Shen (including reference to the 3-armed sweater)	3: Key DevOps Principles	https://www.jasonshen.com/2015/first-impressions-at-etsy/
'From Containers to Microservices: Modernizing Legacy Applications' on TechBeacon	6: Automation & Architecting Toolchains	https://techbeacon.com/containers-microservices-how-modernize-legacy-applications

DevOps Foundation Course: Value Added Resources

'From Darwin to DevOps: John Willis and Gene Kim Talk about Life after The Phoenix Project' by Helen Beal	4: Business & Technology Frameworks	https://www.infoq.com/news/2018/05/Beyond-The-Phoenix-Project
'How to Find Your Continuous Delivery Rhythm' from devops.com	1: Exploring DevOps	http://devops.com/features/find-continuous-delivery-rhythm/
'How to Implement a Solid DevOps Team Structure' by Alex Barrett	8: Sharing, Shadowing & Evolving	http://searchitoperations.techtarget.com/feature/How-to-implement-a-solid-DevOps-team-structure
'How to Reduce Employee Resistance to Change' by Susan. M. Heathfield	5: DevOps Values: Culture, Behaviors & Operating Models	https://www.thebalancecareers.com/how-to-reduce-employee-resistance-to-change-1918992
'Introducing DevOps Where ITIL Rules' by Alan Sharp-Paul	4: Business & Technology Frameworks	http://www.scriptrock.com/blog/introducing-devops-itil-rules-enterprise-2
'Is Yours a Learning Organization?' on HBR	4: Business & Technology Frameworks	https://hbr.org/2008/03/is-yours-a-learning-organization
'Jesse Robbins Discusses DevOps and Cloud Computing' on Thoughtworks' blog	6: Automation & Architecting Toolchains	http://www.thoughtworks.com/insights/blog/jesse-robbins-discusses-devops-and-cloud-computing
'Just What is a DevOps Engineer?' on devops.com	8: Sharing, Shadowing & Evolving	https://dzone.com/articles/just-what-devops-engineer
'Inside Atlassian: how IT & SRE use ChatOps to run incident management' by Sean Regan	3: Key DevOps Principles	https://www.atlassian.com/blog/it-service-management/inside-atlassian-sre-use-chatops-run-incident-management
'Let's Fund Teams Not Projects' from the DEFRA Digital blog, .gov.uk	4: Business & Technology Frameworks	https://defradigital.blog.gov.uk/2017/09/19/lets-fund-teams-not-projects/
'Machine Learning: AI Driving DevOps Evolution' by Tony Bradley	6: Automation & Architecting Toolchains	https://devops.com/machine-learning-ai-driving-devops-evolution/
'Metrics for DevOps Initiatives' from IT Revolution	7: Measurement, Metrics & Reporting	http://devopsentrepreneur.io/media/DOES_forum_metrics_102015.pdf
'Misconceptions About Kanban' by Leon Tranter	3: Key DevOps Principles	https://www.extremeuncertainty.com/misconceptions-about-kanban/
'Quantifying DevOps Capability: It's Important To Keep CALMS' by Rackspace	1: Exploring DevOps	https://blog.rackspace.com/quantifying-devops-capability-its-important-to-keep-calms
'Resilience Engineering' by Erik Hollnagel	3: Key DevOps Principles	http://erikhollnagel.com/ideas/resilience-engineering.html
'T-Shaped Developers are the New Normal' by David Walker	8: Sharing, Shadowing & Evolving	http://www.theregister.co.uk/2015/11/02/t_shaped_developers_are_the_new_normal/

DevOps Foundation Course: Value Added Resources

'The 7 Skills Ops Pros Need to Succeed in DevOps' by George Hulme	8: Sharing, Shadowing & Evolving	https://devops.com/devops-7-skills-ops-pros-need-succeed/
'The Andon Cord' by John Willis on IT Revolution	4: Business & Technology Frameworks	https://itrevolution.com/kata/
'The Biggest Myth in Building Learning Culture' by Shannon Tipton	2: Core DevOps Practices	http://learningrebels.com/2014/09/12/the-biggest-myth-in-building-a-learning-culture/
'The Convergence of DevOps' by John Willis on IT Revolution	4: Business & Technology Frameworks	http://itrevolution.com/the-convergence-of-devops/
'The DevOps Movement Fits Perfectly with ITSM' by Gene Kim	4: Business & Technology Frameworks	https://www.thinktanks.co.za/the-devops-movement-fits-perfectly-with-itsm/
'The Importance of a Learning Culture' by Skillpath	2: Core DevOps Practices	http://www.skillpath.com/whitepapers/pdf/learningculture.pdf
The Scrum Guide™	4: Business & Technology Frameworks	http://www.scrumguides.org/scrum-guide.html
"The Industry Just Can't Decide About DevOps Teams" by Helen Beal	8: Sharing, Shadowing & Evolving	https://www.infoq.com/news/2017/10/devops-teams-good-or-bad
'The Mission of a DevOps Team' by Casey West	8: Sharing, Shadowing & Evolving	http://caseywest.com/the-mission-of-a-devops-team/
'Theory of Constraints' by Lean Production	2: Core DevOps Practices	http://www.leanproduction.com/theory-of-constraints.html
'There's No Such Thing as a DevOps Team' by Jez Humble	8: Sharing, Shadowing & Evolving	https://continuousdelivery.com/2012/10/theres-no-such-thing-as-a-devops-team/
'Top 25 Lean Tools' on Lean Production	4: Business & Technology Frameworks	http://www.leanproduction.com/top-25-lean-tools.html
'Toyota Kata: Habits for Continuous Improvement' by Håkan Forss	4: Business & Technology Frameworks	http://www.methodsandtools.com/archive/tokyotakata.php
'Tracking Every Release' Code as Craft (Etsy)	7: Measurement, Metrics & Reporting	https://codeascraft.com/2010/12/08/tracking-every-release/
'Transforming the Annual Budgeting Process for DevOps' by Mustafa Kapadia	4: Business & Technology Frameworks	https://devops.com/transforming-the-annual-budgeting-process-for-devops/
Understanding DevOps – Part 4: Continuous Testing and Continuous Monitoring	3: Key DevOps Principles	https://sdarchitect.wordpress.com/2012/10/30/understanding-devops-part-4-continuous-testing-and-continuous-monitoring/
Understanding the Kubler-Ross Change Curve - Cleverism	5: DevOps Values: Culture, Behaviors & Operating Models	http://www.cleverism.com/understanding-kubler-ross-change-curve/
'Use DevOps to Turn IT Into a Strategic	1: Exploring DevOps	http://dev2ops.org/2012/09/use-devops-to-

‘Weapon’ by Damon Edwards		turn-it-into-a-strategic-weapon
‘What DevOps Means to Me’ by John Willis	1: Exploring DevOps	https://blog.chef.io/2010/07/16/what-devops-means-to-me/
‘What is Site Reliability Engineering?’ an interview with Niall Murphy and Ben Treynor at Google	3: Key DevOps Principles	https://landing.google.com/sre/interview/ben-treynor.html
‘What Happens to Test in a DevOps World’ on devops.com	3: Key DevOps Principles	http://devops.com/2016/06/13/happens-test-devops-world/
‘What’s Lost With a DevOps Team’ by Michael Nygard	8: Sharing, Shadowing & Evolving	http://www.michaelnygard.com/blog/2016/01/whats-lost-with-a-devops-team/
‘What’s the Best Team Structure for DevOps Success?’ by Alanna Brown	8: Sharing, Shadowing & Evolving	https://puppet.com/blog/what%E2%80%99s-best-team-structure-for-devops-success
‘What’s the Difference Between AI, Machine Learning and Deep Learning?’ by Michael Copeland	6: Automation & Architecting Toolchains	https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/
‘Why DevOps Engineer is the Number One Hardest Tech Job to Fill by Logicworks	8: Sharing, Shadowing & Evolving	https://www.cloudcomputing-news.net/news/2016/jun/16/devops-engineer-hardest-job-to-fill/
‘Why Everyone Needs DevOps Now’ by Gene Kim	2: Core DevOps Practices	http://www.slideshare.net/realgenekim/why-everyone-needs-devops-now

WebSites

Title	Link
Agile Manifesto	http://www.agilemanifesto.org/
Beyond Budgeting	https://bbrt.org/
DevOps Institute	https://devopsinstitute.com/
DevOps Topologies (Matthew Skelton & Manuel Pais)	https://web.devopstopologies.com/
DevOps.com	https://devops.com/
DevOpsDays	https://www.devopsdays.org/
DevSecOps Reference Architectures (Sonatype)	https://www.sonatype.com/devsecops-reference-architectures
Hubot by Github	https://hubot.github.com/
IT Revolution	https://itrevolution.com/

Periodic Table of DevOps Tools (XebiaLabs)	https://xebialabs.com/periodic-table-of-devops-tools/
Principles of Chaos Engineering	https://principlesofchaos.org
Rugged Software	https://www.ruggedsoftware.org/
SAFe	https://www.scaledagileframework.com
Scrum.org	https://www.scrum.org/
Theory of Constraints Institute	https://www.tocinstitute.org/

DevOps & Engineering Blogs

Blog	Link
AirBNB Engineering & Data Science	https://medium.com/airbnb-engineering
Backstage Blog (SoundCloud)	https://developers.soundcloud.com/blog/
BlackRock Blog	http://rockthecode.io/
code.flickr.com	http://code.flickr.net/
DEFRA Digital	https://defradigital.blog.gov.uk/
Deliveroo Engineering Team	https://deliveroo.engineering/
Dropbox Tech Blog	https://blogs.dropbox.com/tech/
eBay Tech Blog	https://www.ebayinc.com/stories/blogs/tech/
Etsy - Code as Craft	https://codeascraft.com/
Eventbrite Engineering	https://www.eventbrite.com/engineering/
Facebook Engineering	https://www.facebook.com/Engineering
Github Engineering	https://githubengineering.com/
Google Developers	https://developers.googleblog.com/
Heroku Engineering	https://blog.heroku.com/engineering
HubSpot Engineering	https://product.hubspot.com/blog/topic/engineering

Instagram Engineering	http://instagram-engineering.tumblr.com/
Kickstarter Engineering	https://kickstarter.engineering/
LinkedIn Engineering	https://engineering.linkedin.com/blog
Monzo Technology	https://monzo.com/blog/technology/
Moonpig Engineering	https://engineering.moonpig.com/
Netflix TechBlog	https://medium.com/netflix-techblog
PayPal Engineering	https://www.paypal-engineering.com/
Pinterest Engineering	https://medium.com/@Pinterest_Engineering
Revolut Engineering	https://blog.revolut.com/tag/engineering/
Salesforce Engineering	https://engineering.salesforce.com/
Slack Engineering	https://slack.engineering/
Target Tech	http://target.github.io/devops/the-dojo
Ticketmaster Technology	https://tech.ticketmaster.com/category/devops/
Trainline Engineering	https://engineering.thetrainline.com/
Uber Engineering	https://eng.uber.com/
Vimeo Engineering	https://medium.com/vimeo-engineering-blog
Zapier Engineering	https://zapier.com/engineering/

GitHub Resources

Item	Link
CapitalOne DevOps Dashboard	https://github.com/capitalone/Hygieia
Chaos Monkey	https://github.com/Netflix/SimianArmy/wiki/Chaos-Monkey
DevOps Against Humanity	https://github.com/bridgetkromhout/devops-against-humanity
DevOps Tools Collection	https://github.com/collections/devops-tools

Additional Videos of Interest

Title	Link

'Continuous Delivery' with Jez Humble (46:59)	https://youtu.be/skLjuksCRTw
'Continuous Delivery is a Team Sport' with Jez Humble, Gene Kim and Gary Gruver hosted by ElectricCloud (5:33)	https://youtu.be/9XmvFvdTObY
'Get Loose! Microservices & Loosely Coupled Architectures' with Jez Humble and Anders Walgreen hosted by devops.com	https://youtu.be/l9BymWx8G1E
'Intro to Scrum in Under 10 Minutes' by Axosoft	https://youtu.be/XU0lIRltyFM
'Learn How Value Stream Mapping Applies to Any Industry or Process" by Gemba Academy	https://youtu.be/gg5u9kn0Bzo
Sidney Dekker, Richard Cook and Stephen Spear at DOES 2017	https://youtu.be/CFMJ3V4VakA
'The Real DevOps of Silicon Valley' from AppDynamics (4:49)	https://youtu.be/2PjVuTzA2Ik

DevOps Books

Title	Author	Link
Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations	Nicole Forsgren PHD, Jez Humble & Gene Kim	https://itrevolution.com/book/accelerate/
Beyond The Phoenix Project	Gene Kim and Jez Humble	https://itrevolution.com/book/beyond-phoenix-project/
Continuous Delivery	Jez Humble and Dave Farley	https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912
DevOps for the Modern Enterprise	Mirco Hering	https://itrevolution.com/book/devops_modern_enterprise/
Just Culture	Sidney Dekker	http://sidneydekker.com/just-culture/
Leading Change	John P Kotter	https://www.amazon.com/Leading-Change-New-Preface-Author-ebook/dp/B00A07FPEO
Lean IT	Steven C Bell and Michael A Orzen	https://www.amazon.com/Lean-Enabling-Sustaining-Your-Transformation/dp/1439817561
Site Reliability Engineering	Niall Richard Murphy, Betsy Beyer and Chris Jones	https://www.amazon.com/Site-Reliability-Engineering-Production-

DevOps Foundation Course: Value Added Resources

		Systems/dp/149192912X
The Art of Business Value	Mark Schwartz	https://itrevolution.com/book/the-art-of-business-value/
The DevOps Handbook	Gene Kim, Jez Humble, Patrick Debois & John Willis	https://itrevolution.com/book/the-devops-handbook/
The Phoenix Project	Kevin Behr, George Spafford and Gene Kim	https://itrevolution.com/book/the-phoenix-project/
The Field Guide to Understanding Human Error	Sidney Dekker	https://www.routledge.com/The-Field-Guide-to-Understanding-Human-Error-3rd-Edition/Dekker/p/book/9781472439055

Case Stories Featured in the Course

Company	Module	Link
Alaska Air	4: Business & Technology Frameworks	<ul style="list-style-type: none"> ● https://blog.chef.io/2017/03/20/delivering-the-continuous-enterprise-with-agile-lean-and-devops-aldo-practices/ ● https://www.youtube.com/watch?v=mGJkhuRlvTo ● https://www.slideshare.net/chef-software/alaska-airlines-devops-journey ● https://www.ca.com/us/collateral/case-studies/alaska-airlines-enables-air-travel-innovation-via-apis.html ● https://azure.microsoft.com/en-us/resources/videos/alaska-airlines-visual-studio-team-services-xamarin/ ● https://hostingjournalist.com/video/alaska-airlines-makes-shopping-easier-with-faster-flow-of-new-e-commerce-features/ ● https://medium.com/@weekstweets/modern-infrastructure-automation-a4afe661cad9 ● https://www.solutionsiq.com/resource/agile-amped-podcast/mobile-programming-at-alaska-airlines-with-agile-amped-at-aatc2016/
Australia Post (in notes on SAFe slide)	4: Business & Technology Frameworks	<ul style="list-style-type: none"> ● https://auspostenterprise.com.au/insights/digitising-services/australia-posts-agile-approach-digital-transformation
Capital One	3: Key DevOps Principles	<ul style="list-style-type: none"> ● https://goatcan.do/2016/06/16/the-goat-farm-episode-13-measuring-success-at-capital-one/ ● https://dzone.com/articles/capital-one-a-devops-powerhouse

DevOps Foundation Course: Value Added Resources

		<ul style="list-style-type: none"> ● https://www.youtube.com/watch?v=Re8UvbMU2tw ● http://www.capitalone.io/blog/hygieia-making-sense-out-of-your-devops-tools/
Disney	8: Sharing, Shadowing & Evolving	<ul style="list-style-type: none"> ● https://thenewstack.io/magic-behind-disney-devops-experience/ ● https://puppet.com/blog/disney-s-devops-journey-a-devops-enterprise-summit-reprise ● https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-Disney-organized-for-a-DevOps-transition ● https://www.computerworlduk.com/devops/systems-strategy-chief-jason-cox-details-disneys-devops-journey-3642785/ ● https://www.infoq.com/news/2017/06/does17-keynotes-day-one
Fannie-Mae	6: Automation & Architecting DevOps Toolchains	<ul style="list-style-type: none"> ● http://searchitoperations.techtarget.com/feature/Fannie-Mae-securitization-app-leads-DevOps-implementation ● https://dzone.com/articles/applying-devops-to-deliver-quality-at-speed
ING Bank	1: Exploring DevOps	<ul style="list-style-type: none"> ● http://www.theregister.co.uk/2016/07/01/ing_mainframe_strategy/?mt=1467618827333 ● https://www.finextra.com/newsarticle/29125/ing-bangs-the-drum-for-devops ● http://www.slideshare.net/CAinc/continuous-delivery-the-ing-story-improving-time-to-market-with-devops-and-continuous-delivery ● https://www.youtube.com/watch?v=9jqY_bvI5vk
Societe Generale	7: Measurement, Metrics & Reporting	<ul style="list-style-type: none"> ● https://blog.xebialabs.com/2016/11/15/reap-rewards-devops-one-banks-story/
Target	5: Culture, Behaviors & Operating Models	<ul style="list-style-type: none"> ● https://www.youtube.com/watch?v=7s-VbB1fG5o&feature=youtu.be
Ticketmaster	2: Core DevOps Practices	<ul style="list-style-type: none"> ● http://www.computerweekly.com/news/450416287/How-to-apply-DevOps-practices-to-legacy-IT



DEVOPS

GLOSSARY OF TERMS

This glossary is provided for reference only as it contains key terms that may or may not be examinable.

DevOps Glossary of Terms

Term	Definition
2-Factor or 2-Step Authentication	Two-Factor Authentication, also known as 2FA or TFA or Two-Step Authentication is when a user provides two authentication factors; usually firstly a password and then a second layer of verification such as a code texted to their device, a shared secret, a physical token or biometrics.
A3 Problem Solving	A structured problem-solving approach that uses a lean tool called the A3 Problem-Solving Report. The term "A3" represents the paper size historically used for the report (a size roughly equivalent to 11" x 17").
Access Management	Granting an authenticated identity access to an authorized resource (e.g., data, service, environment) based on defined criteria (e.g., a mapped role), while preventing an unauthorized identity access to a resource.
Access Provisioning	Access provisioning is the process of coordinating the creation of user accounts, e-mail authorizations in the form of rules and roles, and other tasks such as provisioning of physical resources associated with enabling new users to systems or environments.
Advice Process	<p>Any person making a decision must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business.</p> <p>See: http://reinventingorganizationswiki.com/Decision_Making</p>
Agile (adjective)	<p>Able to move quickly and easily; well-coordinated.</p> <p>Able to think and understand quickly; able to solve problems and have new ideas.</p>
Agile Enterprise	Fast moving, flexible and robust company capable of rapid response to unexpected challenges, events, and opportunities.
Agile Manifesto	A formal proclamation of values and principles to guide an iterative and people-centric approach to software development.
Agile Principles	The twelve principles that underpin the Agile Manifesto.
Agile Process Design	The aspect of Agile Service Management (Agile SM) that applies the same Agile approach to process design as developers do to software development.
Agile Process Improvement	The aspect of Agile SM that aligns Agile values with ITSM processes through continuous improvement.
Agile Process Owner	An ITSM or other type of process owner that uses Agile and Scrum principles and practices to design, manage and measure individual processes.
Agile Service Management	Framework that 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.

Agile Service Manager	The operational equivalent to Dev's ScrumMaster. A role within an IT organization that understands how to leverage Agile and Scrum methods to improve the design, speed and agility of ITSM processes.
Agile Software Development	Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Usually applied using the Scrum or Scaled Agile Framework approach
Amazon Web Services (AWS)	Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
Antifragile	The ability to <i>improve</i> with stress rather than merely resist it.
Application Programming Interface (API)	A set of protocols used to create applications for a specific OS or as an interface between modules or applications.
Architecture	The fundamental underlying design of computer hardware, software or both in combination.
Artifact	Any element in a software development project including documentation, test plans, images, data files and executable modules.
Attack path	The chain of weaknesses a threat may exploit to achieve the attacker's objective. For example, an attack path may start by compromising a user's credentials, which are then used in a vulnerable system to escalate privileges, which in turn is used to access a protected database of information, which is copied out to an attacker's own server(s).
Authentication	The process of verifying an asserted identity. Authentication can be based on what you know (eg, password or PIN), what you have (token or one-time code), what you are (biometrics) or contextual information.
Authorization	The process of granting roles to users to have access to resources.
Basic Security Hygiene	A common set of minimum security practices that must be applied to all environments without exception. Practices include basic network security (firewalls and monitoring), hardening, vulnerability and patch management, logging and monitoring, basic policies and enforcement (may be implemented under a "policies as code" approach), and identity and access management.
Bug	An error in software that results in an unexpected or system-degrading condition.
Burndown Chart	Chart showing the evolution of remaining effort against time.
Business Transformation	Changing how the business functions. Making this a reality means changing culture, processes, and technologies in order to better align everyone around delivering on the organization's mission.
Cadence	Flow or rhythm of events.
CALMS Model	Considered the pillars or values of DevOps: Culture, Automation, Lean, Measurement, Sharing (as put forth by John Willis, Damon Edwards and Jez Humble).
Carrots	Positive incentives, for encouraging and rewarding desired behaviors.
Change	Addition, modification or removal of anything that could have an effect on IT services. (ITIL® definition)

Change Failure Rate	A measure of the percentage of failed/rolled back changes.
Change Fatigue	A general sense of apathy or passive resignation towards organizational changes by individuals or teams.
Change Lead Time	A measure of the time from a request for change to delivery of the change.
Change Management	Process that controls all changes throughout their lifecycle. (ITIL definition)
ChatOps	An approach to managing technical and business operations through a group chat room (coined by GitHub).
Cloud Computing	The practice of using remote servers hosted on the internet to host applications rather than local servers in a private datacenter.
Code Repository	A repository where developers can commit and collaborate on their code. It also tracks historical versions and potentially identifies conflicting versions of the same code. Also referred to as "repository" or "repo."
Collaboration	People jointly working with others towards a common goal.
Configuration Management	From Wikipedia: "Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life." https://en.wikipedia.org/wiki/Configuration_management
Constraint	Limitation or restriction; something that constrains. See also bottleneck.
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.
Continual Service Improvement (CSI)	One of the ITIL Core publications and a stage of the service lifecycle.
Continuous Delivery	A methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.
Continuous Deployment	A set of practices that enable every change that passes automated tests to be automatically deployed to production.
Continuous Flow	Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps.
Continuous Integration	A development practice that requires developers to merge their code into a shared repository — ideally, multiple times per day.
Cooperation vs. Competition	The key cultural value shift toward being highly collaborative and cooperative, and away from internal competitiveness and divisiveness.
Critical Success Factor (CSF)	Something that must happen for an IT service, process, plan, project or other activity to succeed.
CSI Register	Vehicle for recording and managing improvement opportunities throughout their lifecycle.

Culture (Organizational Culture)	The values and behaviors that contribute to the unique psychosocial environment of an organization.
Cycle Time	A measure of the time from start of work to ready for delivery.
Daily Scrum	Daily timeboxed event of 15 minutes or less for the Team to replan the next day of work during a Sprint.
Definition of Done (DoD)	Shared understanding of what it means for work to be complete.
Deming Cycle	A four-stage cycle for process management, attributed to W. Edwards Deming. Also called Plan-Do-Check-Act (PDCA).
Deployment	The installation of a specified version of software to a given environment (e.g., promoting a new build into production).
Dev	Individuals involved in software development activities such as application and software engineers.
DevOps	A cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably." (Source: Wikipedia)
DevOps Toolchain	The tools needed to support a DevOps continuous integration, continuous deployment, and continuous release and operations initiative.
DevSecOps	A 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.
DMZ (network security zone)	A DMZ in network security parlance is a network zone in between the public internet and internal protected resources. Any application, server, or service (including APIs) that need to be exposed externally are typically placed in a DMZ. It is not uncommon to have multiple DMZs in parallel.
Dynamic Application Security Testing (DAST)	A type of testing that runs against built code to test exposed interfaces.
Erickson (Stages of Psychosocial Development)	Erik Erikson (1950, 1963) proposed a psychoanalytic theory of psychosocial development comprising eight stages from infancy to adulthood. During each stage, the person experiences a psychosocial crisis which could have a positive or negative outcome for personality development.
Federated Identity	A central identity used for access to a wide range of applications, systems, and services, but with a particular skew toward web-based applications. Also often referenced as Identity-as-a-Service (IDaaS). Any identity that can be reused across multiple sites, particularly via SAML or OAuth authentication mechanisms.

Flow	How people, products or information move through a process. Relative to information and culture, please see: http://reinventingorganizationswiki.com/Information_Flow
Freedom and Responsibility	A core cultural value that with the freedom of self-management (such as afforded by DevOps) comes the responsibility to be diligent, to follow the advice process and to take ownership of both successes and failures.
Fuzzing	From Wikipedia: "Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program." https://en.wikipedia.org/wiki/Fuzzing
Generativity	A cultural view wherein long-term outcomes are of primary focus, which in turn drives investments and cooperation that enable an organization to achieve those outcomes.
Golden Circle	A model by Simon Sinek that emphasizes an understanding of the business' "why" before focusing on the "what" and "how".
Golden Image	A template for a virtual machine (VM), virtual desktop, server or hard disk drive. (TechTarget)
Governance, Risk Management and Compliance (GRC) (platform/software)	A software platform intended for concentrating governance, compliance and risk management data, including policies, compliance requirements, vulnerability data, and sometimes asset inventory, business continuity plans, etc. In essence, a specialized document and data repository for security governance.
Governance, Risk Management and Compliance (GRC) (practice area)	A team of people who specialize in IT/security governance, risk management and compliance activities. Most often non-technical business analyst resources.
Hardening	Securing a server or infrastructure environment by removing or disabling unnecessary software, updating to known good versions of the operating system, restricting network-level access to only that which is needed, configuring logging in order to capture alerts, configuring appropriate access management and installing appropriate security tools.
High-trust Culture	Organizations with a high-trust culture encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas.
Identity	The unique name of a person, device, or the combination of both that is recognized by a digital system. Also referred to as an "account" or "user."
Identity and Access Management (IAM)	From webopedia: "[A] framework of policies and technologies for ensuring the proper people have the appropriate access to technology resources." (http://www.webopedia.com/TERM/I/iam-identity-and-access-management.html)
Identity as a Service (IDaaS)	Identity and access management services that are offered through the cloud or on a subscription basis.
Impediment (Scrum)	Anything that prevents a team member from performing work as efficiently as possible.
Improvement Kata	A structured way to create a culture of continuous learning and improvement. (In Japanese business, Kata is the idea of doing things the "correct" way. An organization's culture can be characterized as its Kata through its consistent role modeling, teaching and coaching.)

Incentive model	A system designed to motivate people to complete tasks toward achieving objectives. The system may employ either positive or negative consequences for motivation.
Incident	Any unplanned interruption to an IT service or reduction in the quality of an IT service. Includes events that disrupt or could disrupt the service. (ITIL definition)
Incident Management	Process that restores normal service operation as quickly as possible to minimize business impact and ensure that agreed levels of service quality are maintained. (ITIL definition)
Incident Response	"[An] organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way that limits damage and reduces recovery time and costs." http://searchsecurity.techtarget.com/definition/incident-response
Increment	Potentially shippable completed work that is the outcome of a Sprint.
Infrastructure	All of the hardware, software, networks, facilities, etc., required to develop, test, deliver, monitor and control or support IT services. The term IT infrastructure includes all of the information technology but not the associated people, processes and documentation. (ITIL definition)
Infrastructure as Code	The practice of using code (scripts) to configure and manage infrastructure.
Integrated development environment (IDE)	An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. (TechTarget)
Integrated development environment (IDE) 'lint' checks	Linting is the process of running a program that will analyze code for potential errors (e.g., formatting discrepancies, non-adherence to coding standards and conventions, logical errors).
Internet of Things	A network of physical devices that connect to the internet and potentially to each other through web-based wireless services.
INVEST	A mnemonic was created by Bill Wake as a reminder of the characteristics of a quality user story
ISO 31000	A family of standards that provide principles and generic guidelines on risk management.
ISO/IEC 20000	International standard for IT service management. ISO/IEC 20000 is used to audit and certify service management capabilities.
Issue Management	A process for capturing, tracking, and resolving bugs and issues throughout the software development lifecycle.
ITIL	Set of best practice publications for IT service management. Published in a series of five core books representing the stages of the IT service lifecycle which are: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement.
IT Service	A service provided to a customer from an IT organization.

Kaizen	The practice of continuous improvement.
Kanban	Method of work that pulls the flow of work through a process at a manageable pace.
Kanban Board	Tool that helps teams organize, visualize and manage work.
Key Performance Indicator (KPI)	Key metric used to measure the achievement of critical success factors. KPIs underpin critical success factors and are measured as a percentage. (ITIL definition)
Knowledge Management	Process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision.
Knowledge Management	Process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision.
Known Error	Problem with a documented root cause and a workaround. (ITIL definition)
Kubler-Ross Change Curve	Describes and predicts the stages of personal and organizational reaction to major changes.
Laloux (Culture Models)	Frederic Laloux created a model for understanding organizational culture.
Lean (adjective)	Spare, economical. Lacking richness or abundance.
Lean (production)	Production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value.
Lean Enterprise	Organization that strategically applies the key ideas behind lean production across the enterprise.
Lean IT	Applying the key ideas behind lean production to the development and management of IT products and services.
Lean Manufacturing	Lean production philosophy derived mostly from the Toyota Production System.
Lean Thinking	The goal of lean thinking is to create more value for customers with fewer resources and less waste. Waste is considered any activity that does not add value to the process.
Log Management	"The collective processes and policies used to administer and facilitate the generation, transmission, analysis, storage, archiving and ultimate disposal of the large volumes of log data created within an information system." http://searchitoperations.techtarget.com/definition/log-management
Machine Learning	Data analysis that uses algorithms that learn from data.
Many-factor Authentication	The practice of using at least 2 factors for authentication. The two factors can be of the same class.
Mean Time Between Deploys	Used to measure deployment frequency.
Mean Time Between Failures (MTBF)	Average time that a CI or IT service can perform its agreed function without interruption. Often used to measure reliability. Measured from when the CI or service starts working, until the time it fails (uptime). (ITIL definition)

Mean Time to Detect Incidents (MTTD)	Average time required to detect a failed component or device.
Mean time to Discovery	How long a vulnerability or software bug/defect exists before it's identified.
Mean time to Patch	How long it takes to apply patches to environments once a vulnerability has been identified.
Mean Time to Repair (MTTR)	Average time required to repair a failed component or device. MTTR does not include the time required to recover or restore service.
Mean time to Resolution	How long it takes for a production-impacting issue to be resolved.
Mean Time to Restore Service (MTRS)	Used to measure time from when the CI or IT service fails until it is fully restored and delivering its normal functionality (downtime). Often used to measure maintainability. (ITIL definition)
Metric	Something that is measured and reported upon to help manage a process, IT service or activity.
Microservices	A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system.
Minimum Viable Product	Most minimal version of a product that can be released and still provide enough value that people are willing to use it.
Model	Representation of a system, process, IT service, CI, etc. that is used to help understand or predict future behavior. In the context of processes, models represents pre-defined steps for handling specific types of transactions.
Multi-factor Authentication	The practice of using 2 or more factors for authentication. Often used synonymously with 2-factor Authentication.
Open Source	Software that is distributed with its source code so that end user organizations and vendors can modify it for their own purposes.
Operations Management	Function that performs the daily activities needed to deliver and support IT services and the supporting IT infrastructure at the agreed levels. (ITIL)
Ops	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists and service desk analysts.
Orchestration	An approach to building automation that interfaces or "orchestrates" multiple tools together to form a toolchain.
Organizational Change	Efforts to adapt the behavior of humans within an organization to meet new structures, processes or requirements.
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.

Outcome	Intended or actual results.
Non-functional requirements	Requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors or functions (e.g., availability, reliability, maintainability, supportability); qualities of a system.
Patch	A software update designed to address (mitigate/remediate) a bug or weakness.
Patch management	The process of identifying and implementing patches.
Penetration Testing, or Pentesting	From Wikipedia: "an authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data." https://en.wikipedia.org/wiki/Penetration_test It should be noted that pentesting is distinctly different from vulnerability scanning (vuln scanning), though a vuln scan may be performed during a pentest engagement. Sometimes also referred to as "red teaming" or "tiger teaming," though this isn't necessarily correct. Red teams or tiger teams are designed to model specific threats rather than the more broad pentesting objective of looking for as many weaknesses as possible.
Plan-Do-Check-Act (PDCA)	Four-stage cycle for process management, attributed to W. Edwards Deming. Also known as the Deming Cycle.
Policies	Formal documents that define boundaries in terms of what the organization may or may not do as part of its operations.
Policy as Code	The notion that security principles and concepts can be articulated in code (e.g., software, configuration management, automation) to a sufficient degree that the need for an extensive traditional policy framework is greatly reduced. Standards and guidelines should be implemented in code and configuration, automatically enforced and automatically reported-on in terms of compliance, variance or suspected violations.
Post Implementation Review (PIR)	Review that takes place after a change or a project has been implemented that assesses whether the change was successful and opportunities for improvement.
Potentially Shippable Product	Increment of work that is "done" and capable of being released if it makes sense to do so.
Priority	The relative importance of an incident, problem or change; based on impact and urgency. (ITIL definition)
Privileged Access Management (PAM)	Technologies that help organizations provide secured privileged access to critical assets and meet compliance requirements by securing, managing and monitoring privileged accounts and access. (Gartner)
Problem	The underlying cause of one or more incidents. (ITIL definition)
Procedure	Step-by-step instructions that describe how to perform the activities in a process.
Process	Structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs. Elated work activities that take specific inputs and produce specific outputs that are of value to a customer.

Process Owner	Role accountable for the overall quality of a process. May be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations. (ITIL definition)
Process Backlog	Prioritized list of everything that needs to be designed or improved for a process including current and future requirements.
Process Customer	Recipient of a process' output.
Process Owner	Person accountable for the overall quality of a process and owner of the Process Backlog.
Process Planning Meeting	A high level event to define the goals, objectives, inputs, outcomes, activities, stakeholders, tools and other aspects of a process. This meeting is not timeboxed.
Process Supplier	Creator of process input.
Product Backlog (Scrum)	Requirements for a system, expressed as a prioritized list of product backlog items. The product backlog is prioritized by the Product Owner and includes functional, non-functional and technical team-generated requirements.
Product Backlog Refinement	Ongoing process of adding detail, estimates and order to backlog items. Sometimes referred to as Product Backlog grooming.
Product Owner (Scrum)	An individual responsible for maximizing the value of a product and for managing the product backlog.
Release (noun)	Software that is built, tested and deployed into the production environment.
Release Management	Process that manages releases and underpins Continuous Delivery and the Deployment Pipeline.
Release Planning Meeting	Timeboxed event that establishes the goals, risks, features, functionality, delivery date and cost of a release. It also includes prioritizing the Product Backlog.
Reliability	Measure of how long a service, component or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI. (ITIL definition)
Remediation Plan	Plan that determines the actions to take after a failed change or release. (ITIL definition)
Request for Change (RFC)	Formal proposal to make a change. The term RFC is often misused to mean a change record, or the change itself. (ITIL definition)
Resilience	Building an environment or organization that is tolerant to change and incidents.
Return on Investment (ROI)	Difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage.
Risk	<p>Possible event that could cause harm or loss, or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks and managing risks. The probably frequency and probable magnitude of future loss.</p> <p>Pertains to a possible event that could cause harm or loss, or affect an organization's ability to execute or achieve its objectives.</p>
Risk management process	The process by which "risk" is contextualized, assessed, and treated. From ISO 31000: 1) Establish context, 2) Assess risk, 3) Treat risk (remediate, reduce or accept).

Role	<p>Set of responsibilities, activities and authorities granted to a person or team. A role is defined by a process. One person or team may have multiple roles.</p> <p>A set of permissions assigned to a user or group of users to allow a user to perform actions within a system or application.</p>
Role-based Access Control (RBAC)	An approach to restricting system access to authorized users. https://en.wikipedia.org/wiki/Role-based_access_control
Root Cause Analysis (RCA)	Actions take to identify the underlying cause of a problem or incident.
Rugged Development (DevOps)	Rugged Development (DevOps) is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone.
Scaled Agile Framework (SAFE)	A proven, publicly available, framework for applying Lean-Agile principles and practices at an enterprise scale.
Scrum	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 Components	Scrum's roles, events, artifacts and the rules that bind them together.
Scrum Guide	The definition of Scrum concepts and practices, written by Ken Schwaber and Jeff Sutherland.
ScrumMaster	An individual who provides process leadership for Scrum (i.e., ensures Scrum practices are understood and followed) and who supports the Scrum Team by removing impediments.
Scrum Team	A self-organizing, cross-functional team that uses the Scrum framework to deliver products iteratively and incrementally. The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master.
Security as Code	Building security into DevOps tools and practices, making it an essential part of tool chains and workflows. https://www.safaribooksonline.com/library/view/devopssec/9781491971413/ch04.html
Security (Information Security)	Practices intended to protect the confidentiality, integrity and availability of computer system data from those with malicious intentions.
Service	Means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.
Service Design	One of the ITIL Core publications and a stage of the service lifecycle.
Service Catalog	Subset of the Service Portfolio that consists of services that are live or available for deployment. Has two aspects: the Business/Customer Service Catalog (visible to customers) and the Technical/Supporting Service Catalog. (ITIL definition)
Service Desk	Single point of contact between the service provider and the users.

Service Level Agreement (SLA)	Written agreement between an IT service provider and its customer(s) that defines key service targets and responsibilities of both parties.
Service Lifecycle	Structure of the ITIL Core guidance.
Service Management	Set of specialized organizational capabilities for providing value to customers in the form of services. (ITIL definition)
Service Operation	One of the ITIL Core publications and a stage of the service lifecycle.
Service Provider	Organization that supplies services to one or more internal or external customers. (ITIL definition)
Service Strategy	One of the ITIL Core publications and a stage of the service lifecycle.
Service Request	User request for a standard service from an IT service provider. (ITIL definition)
Service Transition	One of the ITIL Core publications and a stage of the service lifecycle.
Shift Left	An approach that strives to build quality into the software development process by incorporating testing early and often. This notion extends to security architecture, hardening images, application security testing, and beyond.
SMART Goals	Specific, measurable, achievable, relevant and time-bound goals.
Software Composition Analysis	A tool that checks for libraries or functions in source code that have known vulnerabilities.
Sprint (Scrum)	A time-boxed iteration of work during which an increment of product functionality is implemented.
Stakeholder	Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers. (ITIL definition)
Standard Change	Pre-approved, low risk change that follows a procedure or work instruction. (ITIL definition)
Static Application Security Testing (SAST)	A type of testing that checks source code for bugs and weaknesses.
Sticks	Negative incentives, for discouraging or punishing undesired behaviors.
Subversion control repository	A repository where developers can commit and collaborate on their code. It also tracks historical versions and potentially identifies conflicting versions of the same code.
Supplier	External (third party) supplier, manufacturer or vendor responsible for supplying goods or services that are required to deliver IT services.
System of Record	<p>From Wikipedia: "A system of record is a data management term for an information storage system that is the authoritative data source for a given data element or piece of information."</p> <p>https://en.wikipedia.org/wiki/System_of_record</p>

Test-driven development	From Wikipedia: "Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only. This is opposed to software development that allows software to be added that is not proven to meet requirements." https://en.wikipedia.org/wiki/Test-driven_development
Test-driven development cycle	<p>From Wikipedia:</p> <ol style="list-style-type: none"> "1. Add a test 2. Run all tests and see if the new test fails 3. Write the code 4. Run tests 5. Refactor code <p>Repeat"</p> <p>https://en.wikipedia.org/wiki/Test-driven_development#Test-driven_development_cycle</p>
The Checkbox Trap	The situation wherein an audit-centric perspective focuses exclusively on "checking the box" on compliance requirements without consideration for overall security objectives.
The Three Ways	Key principles of DevOps – Flow, Feedback, Continuous experimentation and learning.
Theory of Constraints	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.
Thomas Kilmann Inventory (TKI)	Measures a person's behavioral choices under certain conflict situations.
Threat	A bad actor, human or automated, that acts against a system with intent to harm or compromise that system. Sometimes also called a "Threat Agent" or "Threat Actor." https://en.wikipedia.org/wiki/Factor_analysis_of_information_risk
Threat Intelligence	Information pertaining to the nature of a threat or the actions a threat may be known to be perpetrating. May also include "indicators of compromise" related to a given threat's actions, as well as a "course of action" describing how to remediate the given threat action.
Threat Modeling	From Wikipedia: "[A] process by which potential threats can be identified, enumerated, and prioritized – all from a hypothetical attacker's point of view." https://en.wikipedia.org/wiki/Threat_model
Time to Value	Measure of the time it takes for the business to realize value from a feature or service.
Toolchain	A philosophy that involves using an integrated set of complimentary task specific tools to automate an end to end process (vs. a single-vendor solution).
User	Consumer of IT services. Or, the identity asserted during authentication (aka username).
Value Stream	All of the activities to go from a customer request to a delivered product or service.

Value Stream Mapping	Lean tool that depicts the flow of information, materials and work across functional silos with an emphasis on quantifying waste, including time and quality.
Variable Speed IT	An approach where traditional and digital processes co-exist within an organization while moving at their own speed.
Velocity	Measure of the quantity of work done in a pre-defined interval. The amount of work an individual or team can complete in a given amount of time.
Version control tools	Ensure a 'single source of truth' and enable change control and tracking for all production artifacts.
Voice of the Customer (VOC)	A process that captures and analyzes customer requirements and feedback to understand what the customer wants.
Vulnerability	A weakness in a design, system, or application that can be exploited by an attacker.
Vulnerability Intelligence	Information describing a known vulnerability, including affected software by version, relative severity of the vulnerability (for example, does it result in escalation of privileges for user role, or does it cause a denial of service), exploitability of the vulnerability (how easy/hard it is to exploit), and sometimes current rate of exploitation in the wild (is it being actively exploited or is it just theoretical). This information will also often include guidance on what software versions are known to have remediated the described vulnerability.
Vulnerability management	The process of identifying and remediating vulnerabilities.
Waste (Lean Manufacturing)	Any thing or process that does not add value to a product.
Waterfall (Project Management)	Linear and sequential approach to managing software design and development projects in which progress is seen as flowing steadily (and sequentially) downwards (like a waterfall) /
Westrum (Organization Types)	Ron Westrum developed a typology of organizational cultures that includes three types of organizations: Pathological (power-oriented), Bureaucratic (rule-oriented) and Generative (performance-oriented).
Work in Progress (WIP)	Any work that has been started but has not been completed.
Workaround	Temporary way to reduce or eliminate the impact of incidents or problems. May be logged as a known error in the Known Error Database. (ITIL definition)

Licensed For Use Only By: Serge PETIT sergeplace@gmail.com Sep 11 2019 11:42AM