A photograph of two men in an office setting. One man, wearing glasses and a beard, is in the foreground looking down at something. The other man, younger and without glasses, is behind him looking towards the camera.

DevOps Foundation®

DevOps Foundation R3.1

Welcome to DevOps Foundation

- The purpose of this course is to ensure a common understanding of DevOps goals, business value, vocabulary, concepts and practices

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?

2

- Be brief but don't rush!
- Are you Development, IT Operations or a bit of both? Or something else?
- What's your experience with DevOps? Agile? Lean? ITSM?
- What do you think DevOps is all about?
- What do you want to leave with after class tomorrow?

DevOps Foundation Course Goals

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



Pass the DevOps Foundation Exam

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

3

Key concepts:

- The exam is accredited by the DevOps Institute (devopsinstitute.com) which is a collaborative effort between recognized and experienced leaders in the DevOps, InfoSec and ITSM space
- The DevOps Institute is a learning community for DevOps practices
- Bloom level – this exam has both Bloom 1 and Bloom 2 level questions
 - We'll discuss Bloom's Taxonomy in greater detail on the next slide

About Bloom's Taxonomy



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

4

Key concepts:

- Briefly discuss Bloom's taxonomy
- Benjamin Bloom was an American educational psychologist
- Simply put, Bloom's Taxonomy helps teachers categorize learning objectives and, from there, assess learning achievements
- The DevOps Foundation exam will contain Bloom 1 questions that test learning participants' knowledge of DevOps concepts
- The exam will also contain Bloom 2 questions that test their comprehension of these concepts
- Bloom 2 questions are more example oriented, and require that learners think about what they've learned and demonstrate their understanding

About DevOps Institute



The DevOps Institute 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.

5

- DOI was launched in 2015
- Registered Education partners now span 21 countries

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

6

DOI was launched in 2015. Registered Education Partners now span 21 countries.

Module 1

EXPLORING DEVOPS

Module 1: Exploring DevOps

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

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

Defining DevOps

Module 1: Exploring DevOps

9

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.

Key Concepts:

- DevOps does not have a single source of knowledge unlike other IT frameworks.
- It is being built on the collective experiences and insights of multiple thought leaders and resources including conferences, books, articles, webinars, videos and enterprise experience.
- This is particularly important because the DevOps touches virtually every aspect of IT – culture, automation, management, quality, HR and organizational structure.
- That is just too difficult to encapsulate into a single body of knowledge.

The DevOps Collective Body of Knowledge (2)



11

Key Concepts:

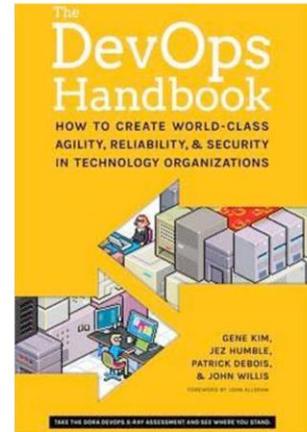
- DevOps does not have a single source of knowledge unlike other IT frameworks
- It is being built on the collective experiences and insights of multiple thought leaders and resources including conferences, books, articles, webinars, videos and enterprise experience
- This is particularly important because the DevOps touches virtually every aspect of IT – culture, automation, management, quality, HR and organizational structure
- That is just too difficult to encapsulate into a single body of knowledge



<https://youtu.be/o7-luYS0iSE>

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."



Module 1: Exploring DevOps

13

Key Concepts:

- This is a definition of DevOps from the introduction on the DevOps handbook
- Note the different stakeholders listed
- Think about all these people sharing a common goal
- Note the concept of fast flow
- Think about what 'planned work' means
- Has it always been possible to balance stability, reliability, availability and security in the IT industry? How well are you doing that today?

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

Mark Schwartz
‘The Art of Business Value’



Module 1: Exploring DevOps

14

Key Concepts:

- Published by IT Revolution (and foreworded by Gene Kim) is Mark Schwartz's 'The Art of Business Value'
- The book puts particular focus on how business value is measured during Agile software projects
- Here Mark refers to 'cycle time' – this might be a metric that matters to you. But the real metric is 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/>

Key Concepts:

- While there are increasing job posting for DevOps Engineers and that the number of DevOps teams is on the rise, DevOps is not a title or team – DevOps is not one person's job, but everyone's job
- Both deviate from the intent of DevOps which is to improve communication and collaboration by creating yet another silo

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

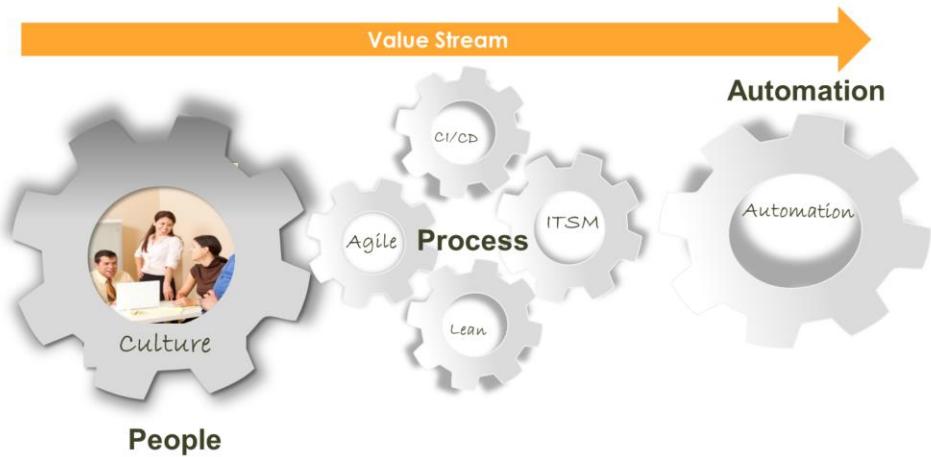
To meet these changing conditions, IT must adapt its culture, practices and automation to be more “continuous”.

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

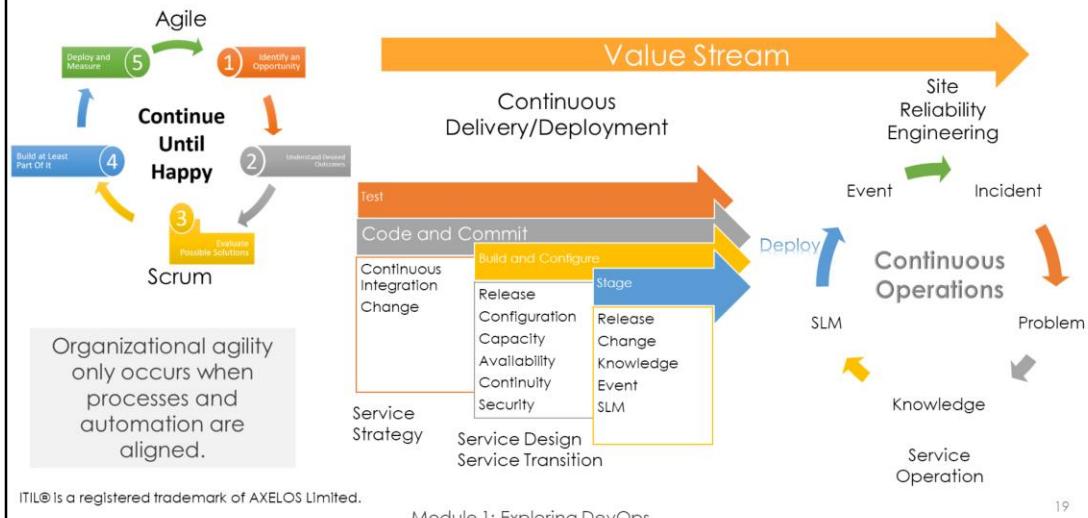


iTIL® is a registered trademark of AXELOS Limited.

Module 1: Exploring DevOps

18

IT is a System of Systems



ITIL® is a registered trademark of AXELOS Limited.

Module 1: Exploring DevOps

19

DISCUSSION

DevOps Myths versus Realities

Module 1: Exploring DevOps

Myths and anti-patterns you may wish to consider are:

- DevOps means chaos
 - DevOps is not the wild west where developers can do whatever they want whenever they want
- DevOps means we don't need IT Operations any more (NoOps)
 - The term 'NoOps' was initially coined by Adrian Cockcroft, Director of Cloud Systems Architecture for Netflix, to describe his IT organization as having little need for operations staff, partly because the company has shifted to the cloud, where it can automate many former functions of the staff. While some organizations strive to entirely replace Ops with automation, there is a recognition that while the role of Ops is evolving there is still a need for operational roles. The article sparked a great deal of debate but Ops' responsibilities are much broader than those only related to software deployment and the concept of NoOps has mostly gone away now -
http://www.pcworld.com/article/252264/noops_debate_grows_heated.html
- DevOps can't be done with legacy systems

- We can't do DevOps if we have outsourced some of our IT capabilities
- Automate everything
- DevOps is only for start-ups (see DevOps Handbook)
- DevOps replaces Agile (see DevOps Handbook)
- DevOps is incompatible with ITIL (see DevOps Handbook)
- DevOps is incompatible with Information Security & Compliance (see DevOps Handbook)
- DevOps is just 'Infrastructure as code' or 'Automation' (see DevOps Handbook)
- DevOps is only for Open Source Software (see DevOps Handbook)

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

Key Concepts:

- Briefly discuss goals
- Many of these goals are interrelated (e.g., smaller, more frequent releases enable organizations to reduce risk, speed up flow and improve quality)
- Time to market is the amount of time between an idea for a product and its availability – which is more important; time to market or time to value?

DevOps Values

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



Module 1: Exploring DevOps

22

Key Concepts:

- After the first US-based DevOps Days in Mountainview 2010 John Willis and Damon Edwards coined the acronym CAMS, which stands for Culture, Automation, Measurement and Sharing
- Jez Humble later added an L, standing for Lean, to form CALMS
- **Culture** – Culture relates to the people and processes aspects of DevOps and the need to improve communication and collaboration
- Without the right culture, automation attempts will be fruitless
- **Automation** – Tools such as release management, configuration management, and monitoring and control tools that enable automation are important aspects of DevOps
- **Lean** – Maximizing customer value while minimizing waste and improving flow
- **Measurement** – If you can't measure, you can't improve – A successful DevOps implementation will measure everything – people, process and technology performance
- **Sharing** – Sharing is the feedback loopback in the CAMS cycle – creating a culture where people share ideas and problems is critical not only because it enables improved communication and collaboration but also because it helps organizations to improve

Additional resources:

- <http://www.getchef.com/blog/2010/07/16/what-devops-means-to-me/>
- <http://devops.com/2015/12/09/devops-and-it-support-4-principles-to-keep-your-team-ahead-of-the-curve/>

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.

Key concepts:

- Automation and Toolchains help increase the speed and consistency of IT delivery
- One of the greatest challenges for enterprises moving toward DevOps is how diverse the IT environments in large organizations tend to be
- It is one of the reasons enterprises have a hard time achieving all of the items in red text at the top of the screen
- There is so much variance between teams in terms of the tools they use, as well as the varying frameworks, methodologies, and processes
- This inconsistency across the enterprise is the enemy of broad, enterprise-wide DevOps success.
- For practices such as continuous integration, continuous delivery, and DevOps to thrive – architects, developers, QA, testers and other teams all need to be using shared systems
- Where appropriate, tie this back to ITSM and roles such as Agile Service Manager and Agile Process Owner since those roles will help adapt underpinning processes to increases in automation.

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:

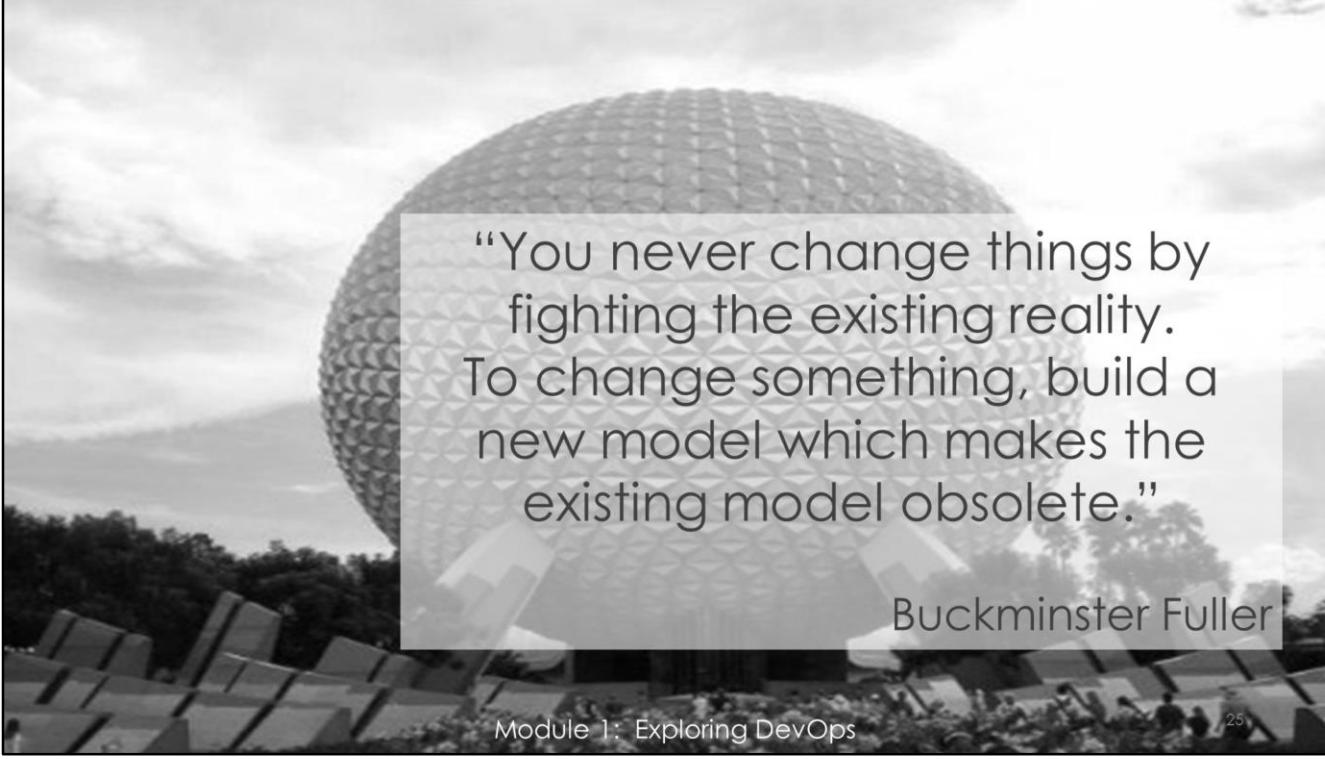
- Information security professionals, systems engineers, system administrators, IT operations engineers, release engineers, database administrators (DBAs), network engineers, support professionals, suppliers, etc.

DevOps extends beyond software developers and IT operations.



Key concepts:

- DevOps is not just about Dev and Ops
- It is about people in other parts of the organization and even outside of the organization working together to improve service delivery



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

Buckminster Fuller

Module 1: Exploring DevOps

25

Key concepts:

- Buckminster Fuller codified the math behind the geodesic dome
- The sentiment behind his quote is very relevant to DevOps

Why Does DevOps Matter?

Module 1: Exploring DevOps

Our Cadence is Off

Historically...



Cadence – the flow or rhythm of events.

Key concepts:

- Businesses want to move quickly and we are all familiar with the mantra 'time is money'
- Technology is only a competitive advantage for a brief period of time and companies must constantly innovate to succeed
- Traditionally, software and application development projects were massive efforts that took 18 to 24 months
- Traditional waterfall projects demand a linear approach that includes requirements definition, design, development, testing and implementation activities wherein one activity can't begin until the previous activity is complete
- This linear approach means that customer feedback is delayed until the project is complete
- Ops, particularly in heavily regulated environments can be just as slow, often due to overly-rigorous methodologies, frameworks and standards coupled with bureaucracy
- Many organizations struggle to do releases every nine months
- Another factor that slows work into Ops is a limited release windows, often over weekends that are time boxed
- All projects must fit into these release windows and if a change fails, the project is further delayed until the next release window
- Pushing all projects into these limited release windows results in complex and turbulent installs

DevOps Improves IT's Cadence and Velocity

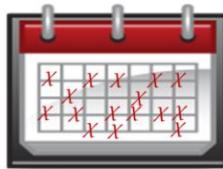
...agile, lean and ITSM practices are also needed.

The Business



Winning through
Innovation

Agile/Lean DevOps



Continuous Delivery



What is the difference between cadence and velocity?

Key concepts:

- Ever higher business expectations mean there has never been a greater need for agility in both Dev and Ops
- Both organizations need to look for ways to reduce waste and to streamline existing processes
- Both organizations need to move more quickly where it is appropriate, that is, where the risks can be mitigated
- Continuous delivery ensures that software is always in a releasable state which allows organizations to introduce incremental changes in smaller batches
 - Continuous delivery is discussed in greater detail later in this course
- There will still be some complex projects that require greater rigor and will need to move more slowly
 - Typically those projects where the requirements are extremely well defined and there is little value in using an incremental approach
- Even when traditional waterfall practices are used, however, projects cannot move so slow that it cripples the business
- IT service management must be able to accommodate either approach by using, for example, change models and more standard changes that can be implemented during normal business hours without disruption

Resource:

<http://devops.com/features/find-continuous-delivery-rhythm/>

DevOps Improves Throughput AND Stability

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

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

"Our research continues to show that the industry-standard Four Key Metrics of software development and delivery drive organizational performance in technology transformations. This year's report revalidates previous findings that it is possible to optimize for stability without sacrificing speed."

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



29

Key concepts:

For 5 years from 2013 - 2018 IT Revolution, DORA and Puppet Labs conducted an extensive survey of practitioners and others on emerging practices in DevOps. In 2018, DORA announced a change in partnership from Puppet to Google Cloud Platform (GCP). The State of DevOps Reports teach us a wealth of things, including:

- It's possible to go faster without compromising quality – there is no necessary trade off between stability and throughput
- Part of that stems from efforts to capture quality at the source and ensure that errors aren't known until passed downstream
- These efforts are accomplished via practices such as version control, continuous integration (discussed shortly), continuous delivery and automated testing
- Another consideration is the fact that organizations are introducing more sophisticated monitoring early in the development lifecycle
- Note: we will be exploring transformational leadership in more detail later on in the course and address it in great depth in the DevOps Leader course

Additional Resources:

- 2019 Accelerate State of DevOps Report:
<https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-report>

[devops-elite-performance-productivity-and-scaling](#)

- DORA/Google collaboration announcement:
<https://www.infoq.com/news/2018/04/DORA-Google-Cloud-New-Research>
- 2018 Accelerate State of DevOps Report: <https://cloudplatformonline.com/2018-state-of-devops.html>
- 2017 State of DevOps Report: <https://puppet.com/resources/whitepaper/state-of-devops-report>
- 2016 State of DevOps Report: <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>
- 2015 State of DevOps Report: <https://puppet.com/resources/whitepaper/2015-state-devops-report>
- 2014 State of DevOps Report: <https://puppet.com/resources/whitepaper/2014-state-devops-report>
- 2013 State of DevOps Report: <https://puppet.com/resources/whitepaper/2013-state-devops-report>

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



- 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

DevOps Adoption



GitLab 2019
Developer
Survey



2018: The
Year Of
Enterprise
DevOps



puppet 2019 State
of DevOps
Report

- 89% more likely to have good insight into what their colleagues are working on
- 45% report continuous code deployment
- 50% say security vulnerabilities are discovered by the security team after code is merged
- 49% encounter the most delays during testing

- 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

- 79% of respondents at 'Medium' level evolution
- 35% of level 5 organizations deploy on demand: 61% could
- 7% can remediate a critical vulnerability within 1 hour
- Organizations at the highest level of DevOps adoption also have fully integrated security practices

Module 1: Exploring DevOps

31

Key concepts:

- Sources vary in their interpretation and gathering of DevOps adoption data, but most organizations are now at least talking about it, if not implementing capabilities that they describe as DevOps
- Later in the course, we'll plot your organization's current capability on an adoption curve

Resources

- <https://about.gitlab.com/developer-survey/>
- <https://go.forrester.com/blogs/2018-the-year-of-enterprise-devops/>
- <https://puppet.com/resources/whitepaper/state-of-devops-report>



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

Damon Edwards

Module 1: Exploring DevOps

Resource:

- <http://dev2ops.org/2012/09/use-devops-to-turn-it-into-a-strategic-weapon/#more-329>

The Business Perspective

Module 1: Exploring DevOps

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.

Statistics

Module 1: Exploring DevOps

POWER C
POWER C
EMAIL
INTERNET
WORLD
NO
34 SEARCH

Key concepts:

- Who is currently undertaking a DevOps initiative?
- What are you doing to get started?

While there is no specific body of knowledge, the very concept of DevOps is encouraging their organization to engage in dialogue, analysis, sharing as the first steps. That is more important than a prescriptive guidance.

The Business Value of DevOps

Combined, commercial and non-commercial goals include:

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



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

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

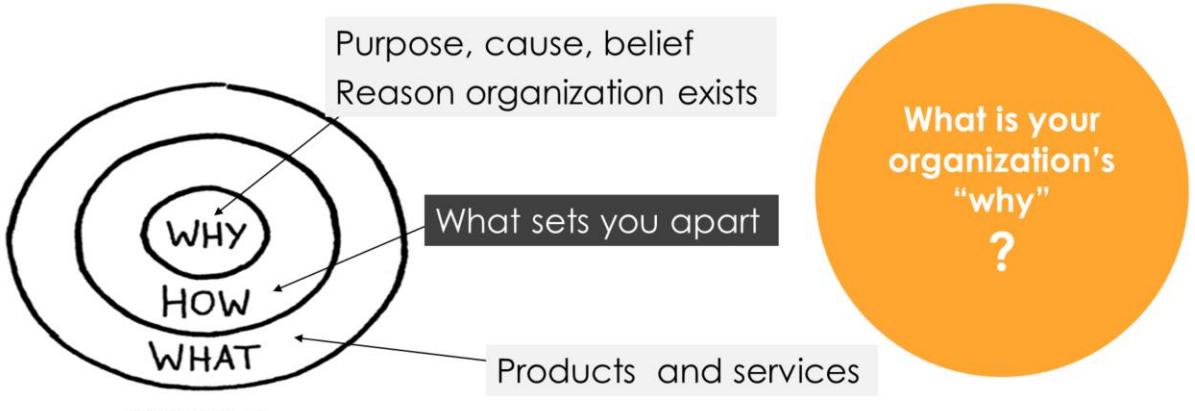
Key concepts:

- From the 2019 State of DevOps Survey Report
 - IT performance strongly correlates with well-known DevOps practices such as use of version control and continuous delivery
 - The longer an organization has implemented — and continues to improve upon — DevOps practices, the better it performs
 - Better IT performance correlates to higher performance for the entire organization
- Organizational culture is one of the strongest predictors of both IT performance and overall performance of the organization
- In the public sector, considerations might include mission effectiveness, cost efficiency, accountability to the public, customer satisfaction, protection of national security

Resources:

Private vs. public sector metrics – <https://balancedscorecard.org/Resources/KPI-Basics/Translating-Metrics>

Start with the “Why” - The Golden Circle



Key concepts:

- In 2009, Simon Sinek gave the talk “How great leaders inspire action” to an audience of 50 people in a small room at TEDxPugetSound
- It quickly became the second most-watched talk on TED.com
- In the context of DevOps, it is critical that any endeavor supports the business’ ‘Why’



<https://www.youtube.com/watch?v=IPYeCltXpxw&feature=youtu.be>

EXERCISE

Your Organizational Why

Module 1: Exploring DevOps

Articulate your organization's purpose in as few words as possible. Example organizational why's include:

- “Think Different.” – Apple
- “Helping Britain Prosper.” – Lloyds Banking Group
- “We exist to make the lives of retired people better.” – Saga
- “We connect people and strengthen relationships.” – Greetings card manufacturer from Karen Martin’s ‘Clarity First’
- “Nourishing families so they can flourish and thrive.” – Kellogg from Karen Martin’s ‘Clarity First’
- “Empowering people to stay a step ahead in life and business.” – ING Bank from Karen Martin’s ‘Clarity First’

Peter Drucker said: “The purpose of a business is to create a customer.”

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

Clyde Logue
Founder of StreamStep



The IT Perspective

Module 1: Exploring DevOps

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?

Module 1: Exploring DevOps

41

Key concepts:

- These are the same drivers as we saw the business being concerned with
- Which matter more to tech than business do you think?
- Do your business AND technology teams recognize all the statements to be true or is there divergence?

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.

Module 1: Exploring DevOps

42

Key concepts:

IT has been taking steps to improve – however they did not do this together. As a result, each initiative gained some benefits but did not show an improvement in the end to end value stream.

Ask the class to give you some examples of the benefits that Agile software development, ITSM and new automation gave their organization. Ask them also for some of the challenges that these initiatives presented and how/if they were able to overcome them.

The Wall of Confusion (1)

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



Module 1: Exploring DevOps

43

Key concepts:

- DevOps is a response to the growing awareness that there is a disconnect between what is traditionally considered development activity and what is traditionally considered operations activity
- This disconnect often manifests itself as conflict and inefficiency
- Dev by its very nature is required and incented to generate change
- Ops is required and incented to maintain stability
 - Mainly because the business has no tolerance for unavailability
- A wall of confusion occurs as a result of these conflicting goals
- This inherent conflict creates a downward spiral that creates slower feature time to market, longer deployment cycles, increasing numbers of outages, and an ever-increasing amount of technical debt

The Wall of Confusion (2)



Key concepts:

- Briefly discuss the characteristics of each end of the spectrum
- An extreme focus on either end of the spectrum is not good
- Some businesses have regulatory controls in place that don't let them take risks (and so they can't always change as quickly as they would like)
- Other organizations are willing to take risks (e.g., startup companies) as it's more important to them to demonstrate innovation and be first to market
- Regardless of an organization's risk tolerance, stability is important
- The business wants both
- DevOps breaks the traditional 'iron triangle' where you can only have software delivered at high speed, low cost with low quality or at low cost, high quality at low speed or at high quality, high speed with high cost – with DevOps you 'can have it all' – software delivered at high speed, high quality and at low cost (with an extra dose of happiness/delight)

IT's Silo Culture



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.

Key concepts:

- What characteristics do developers have?
- What about IT Operations staff?
- And testers?
- And QA?
- And the business?

Stereotypes and perceptions are having a negative cultural impact on the ability for Dev/Ops/Security/Support to interact and collaborate.

Resources:

- The Real DevOps of Silicon Valley from AppDynamics:
<https://www.youtube.com/watch?v=2PjVuTzA2Ik>

Module One Quiz

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

Module 1: Exploring DevOps

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) 20%
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 |

Module 1: Exploring DevOps

Module 2

CORE DEVOPS PRINCIPLES

Module 2: Core DevOps Principles

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

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

The Three Ways

Module 2: Core DevOps Principles

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

Key Concepts:

- The 'Three Ways' are introduced in 'The Phoenix Project: A Novel About It, DevOps, And Helping Your Business Win' by Gene Kim, Kevin Behr and George Spafford
- DevOps practices consistently found in high-performing organizations are emerging and evolving
- The practices that follow align with the principles of the The Three Ways

Resource:

- <http://itrevolution.com/a-personal-reinterpretation-of-the-three-ways/>



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

Module 12: Core DevOps Principles

52

<https://youtu.be/nUOXDEvplRc>

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.

Key Concepts:

- The First Way
 - Emphasizes the performance of the entire system (vs. a single silo or department)
 - Focuses on all business value streams enabled by IT
- In high-performing organizations you'll see lots of examples of flow such as process and value stream maps
- Also in high-performing organizations people self-organize to improve that flow

Resources:

<http://www.slideshare.net/realgenekim/why-everyone-needs-devops-now>

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.

Key Concepts:

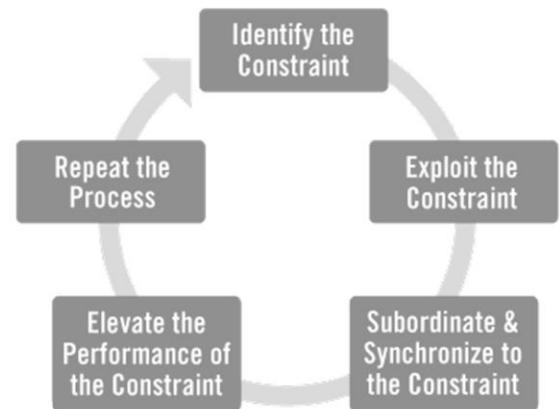
- The Theory of Constraints was first introduced in The Goal, which is a business novel and so is easy to read
- It speaks to the fact that every process has a point that has the potential to be a bottleneck
- To function more efficiently organizations must identify their constraints and widen or eliminate the bottleneck
- Note that not all constraints can be eliminated (e.g., regulatory controls)
- The Phoenix Project was written as a homage to The Goal – Gene Kim explained in Beyond the Phoenix Project: "Our hope was that, in The Phoenix Project, we could describe in equal clarity every sort of problem that every functional silo in the technology value stream also faced."

Resources:

- <http://www.leanproduction.com/theory-of-constraints.html>
- <http://www.tocinstitute.org/theory-of-constraints.html>

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



Key Concepts:

- DevOps is the progression of the software development lifecycle (SDLC) from Waterfall to Agile to Lean
- DevOps leverages Lean by removing waste from the SDLC
- Often, the waste or bottlenecks are found in the form of the constraints shown on this slide
- Examples include inconsistent environments, manual build and deployment processes, poor quality and testing practices, lack of communication and understanding between IT silos, frequent outages and failing SLAs
- All of these issues require IT resources to spend significant time and money keeping the systems running

Resources:

- Get Loose – a webinar with Jez Humble about loosely coupled architectures: <https://youtu.be/l9BymWx8G1E>
- Overly tight architecture – see 10:27 mark of video - <https://www.youtube.com/watch?v=-HSSGiYXA7U> - DOES14 - Gary Gruver - Macy's - Transforming Traditional Enterprise Software Development Processes

DISCUSSION

Overcoming Constraints

Module 2: Core DevOps Principles

Pick a constraint from the previous slide that you find inhibits you frequently, explore its causes and come up with some ideas, based on what you have learned so far, of ways to address it.

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.

Key Concepts:

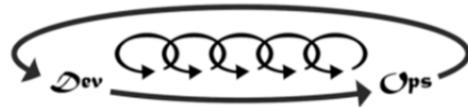
- Frequently when releasing software, information flows from left to right – i.e., from Dev to QA to Ops
- It is less common, however, for right to left feedback – and little time is allocated to reacting to feedback when it is received
- The second way stresses the importance of right to left feedback
- In order for this to be successful, the feedback loops should be short so that information is received in a timely manner
- Feedback should also be visible (e.g., via dashboards) to amplify the feedback
- Without this feedback, teams lack the information needed to continually improve
- With shorter feedback loops
 - Sharing and knowledge grow
 - Trust grows
 - We better understand our customers, both internal and external
 - We fix defects faster and are better able to prevent problems
 - Processes are improved resulting in better flow, faster delivery speeds and less reactive, unplanned work

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)

Key Concepts:

- We need both aspects of The Third Way equally
- Experimentation and taking risks are what ensures that we keep pushing to improve, even if it means going deeper into the danger zone than we've ever gone
- We also need mastery of the skills that can help us retreat out of the danger zone when we've gone too far
- Companies that really want to show their commitment to innovation, and fearlessness when it comes to failure, can't punish people who fail
- The only reason to punish someone because an innovation project failed is carelessness or laziness

Real-World Examples:

- P&G's feminine care division gives a "Presidents Fail Forward Award" to the "team or individual that enabled the organization to significantly learn from a failure"
- Etsy have a '3 Armed Sweater' <https://www.linkedin.com/pulse/first-impressions-etsy-jason-shen/>
- This type of ritual rewards taking risks and sends a message about what company values

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

Key Concepts:

- The Chaos Monkey concept can be used by mature organizations learn from failure
- Chaos Monkey is a service which identifies groups of systems and randomly terminates one of the systems in a group
- The service operates at a controlled time (does not run on weekends and holidays) and interval (only operates during business hours)
- Chaos Monkey only runs in business hours with the intent that engineers will be alert and able to respond

Resource:

- <https://github.com/Netflix/SimianArmy/wiki/Chaos-Monkey>
- <https://principlesofchaos.org/>

CASE STORY: Ticketmaster

"Over the years, we have been extracting these legacy technologies by adding APIs to modernize the interface to our ticketing engines and platforms. We wanted to get them out quickly. To do that, we need to touch a lot of systems. This has driven us to DevOps. For us, it really started with DevOps. Part of our transformation was to focus on delivering business value faster and delivering more of it, and the driver was speed to market of product."

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



Justin Dean, VP
TechOps



Benefits

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

Module 2: Core DevOps Principles

61

<http://www.computerweekly.com/news/450416287/How-to-apply-DevOps-practices-to-legacy-IT>

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’

Key Concepts:

- Creating a learning culture is critical to sustaining growth and innovation
- Training plans helps to identify necessary competencies and encourages common vocabulary and understanding
- Learning can't be something that only occurs in a classroom or that gets cancelled when the budgets are slashed
- When that happens organizations get stuck in the status quo
- Note that at the end of Beyond The Phoenix Project, Gene Kim and John Willis conclude that high performing organizations practicing DevOps principles can be referred to as ‘Dynamic Learning Organizations’

Resource:

- <http://learningrebels.com/2014/09/12/the-biggest-myth-in-building-a-learning-culture/>

EXERCISE

Bringing the Three Ways to Life

Module 2: Core DevOps Principles

- Using a flip chart:
- List each of The Three Ways:
 - Flow
 - Feedback
 - Continuous experimentation and learning
- Identify improvements you could make in your organization for each of the principles

Module Two Quiz

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

Module 2: Core DevOps Principles

64

Module Two Quiz Answers

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

Module 2: Core DevOps Principles

65

Module 3

KEY DEVOPS PRACTICES

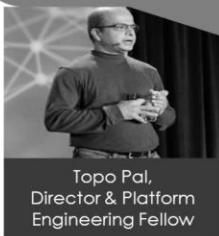
Module 3: Key DevOps Practices

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

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

CASE STORY: Capital One

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



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

Benefits

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



Module 3: Key DevOps Practices

68

- <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>
- <https://www.youtube.com/watch?v=Re8UvbMU2tw>
- <http://www.capitalone.io/blog/hygieia-making-sense-out-of-your-devops-tools/>

Continuous:

- Testing
- Integration
- Delivery
- Deployment

Module 3: Key DevOps Practices

Continuous Testing

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



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

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

Module 3: Key DevOps Practices

70

Key concepts

- The sequence of testing and the need for automated and continuous testing is critical to DevOps; test plans and automation are essential.
- Spend some time discussing functional vs. non-functional requirements
- While the term non-functional may seem to be less important, these are the requirements that underpin the product and must be tested in development and beyond
- Pay particular attention to the rising concept of "shift left" where testing, security, compliance and other functional and non-functional requirements are tested in during the development processes.

Resources:

- <https://sdarchitect.wordpress.com/2012/10/30/understanding-devops-part-4-continuous-testing-and-continuous-monitoring/>
- <http://devops.com/2016/06/13/happens-test-devops-world>

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.

Key Concepts:

- A version control repository is 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.
- A first step towards building quality in at the source is continuous integration
- Is a strong influence on “shift left”
- Continuous integration refers to the continuous integration of multiple code branches into trunk (also known as master) and ensuring that it passes unit tests
- In the context of continuous delivery and DevOps, continuous integration also mandates running on production-like environments, and passing acceptance and integration tests
- This practice makes it possible to detect when code changes break the system as early as possible
- The practice also highlights who caused the break to allow for quick remediation before moving to production
- **IMPORTANT!** Waterfall methodologies can take advantage of continuous integration and test-driven development practices

Resources:

- <http://www.jedi.be/blog/2012/05/12/codifying-devops-area-practices/>

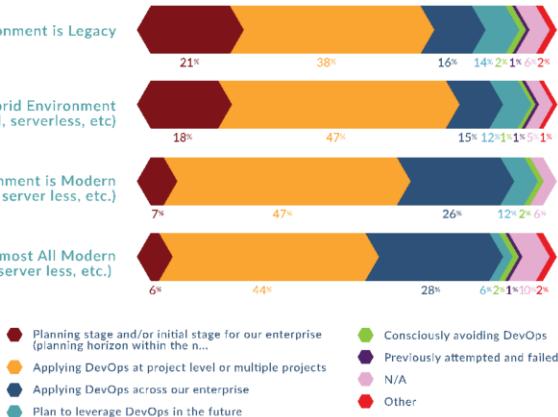
- <https://notafactoryanymore.com/2014/08/19/continuous-everything-in-devops-what-is-the-difference-between-ci-cdcd/>
- <https://www.thoughtworks.com/continuous-integration>

All Environments Leverage DevOps

2019

Upskilling: Enterprise DevOps Skills Report

Presented by DevOps Institute



Module 3: Key DevOps Practices

72

Key Concepts:

- This is from the 2019 Upskilling: Enterprise DevOps Skills Report from DevOps Institute
- The question respondents were asked was: How would you describe the current state of your IT environment? NOTE: independent of what you are moving towards
- DOI did not ask the question of DevOps success as we believe that measuring this is difficult, since those who own the initiative could bias the results
- The real success is in understanding the company's customer experience, which is influenced heavily by the quality and speed in which products and services are delivered
- Let's remember that the key purpose of adopting DevOps is to improve agility, speed and quality of software and service delivery to achieve better customer, employee and partner experience

Additional Resource:

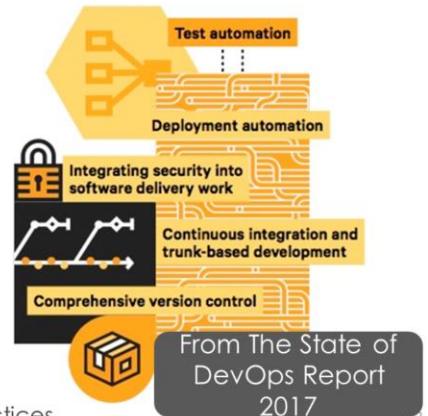
- <https://devopsinstitute.com/2019/03/04/devops-institute-announces-the-2019-upskilling-enterprise-devops-skills-report/>

Continuous Delivery

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

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

Factors that positively contribute to continuous delivery:



Module 3: Key DevOps Practices

Key Concepts:

- Continuous delivery is achieved by continuously integrating the software done by the development team, building executables, and running automated tests on those executables to detect problems
- Furthermore you push the executables into increasingly production-like environments to ensure the software will work in production
- You are doing continuous delivery when:
 - Your software is deployable throughout its lifecycle
 - Your team prioritizes keeping the software deployable over working on new features
 - Anybody can get fast, automated feedback on the production readiness of their systems any time somebody makes a change to them
 - You can perform push-button deployments of any version of the software to any environment on demand
- Benefits of continuous delivery include reduced risk, demonstrable progress and quicker access to user feedback
- How do you differentiate between 'release' and 'deploy' in your organization?

Resources:

- <http://www.informit.com/articles/article.aspx?p=1621865&seqNum=8>
- <https://martinfowler.com/bliki/ContinuousDelivery.html>

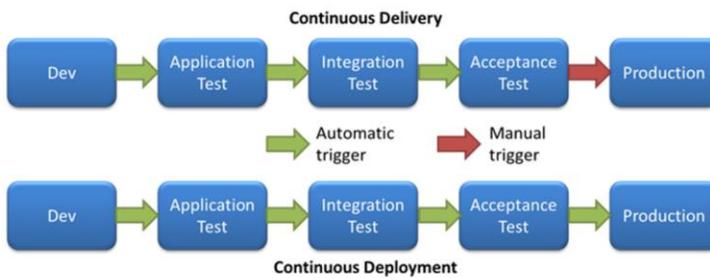
EXERCISE

Rate Your CI/CD Capability

Based on the definitions in the previous slides, rate your capability on the following attributes:

- Developers commit code to master/trunk at least once daily
- A build engine automates a package build process (maybe you use a tool like VSTS, TeamCity, Jenkins or Bamboo)
- Unit tests are performed automatically on commit
- Integration tests are performed automatically on commit
- User acceptance tests are performed automatically on commit (maybe you use a tool like Selenium)
- You have consistent coding standards (maybe you use a tool like Resharper)

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'

Module 3: Key DevOps Practices

75

Key concepts:

- Continuous delivery is sometimes confused with continuous deployment
- Continuous delivery means that you are *able* to do frequent deployments but may choose not to do it, usually due to businesses preferring a slower rate of deployment (typically achieved by batching changes into releases)
- Continuous delivery requires that whenever anyone makes a change that causes an automated test to fail, breaking the deployment pipeline, developers stop the line and bring the system back into a deployable state
- Organizations may even choose to have the continuous integration and deployment system reject any changes (e.g., code or environment commits) that take the code out of a deployable state
- Doing this continually and throughout a development project eliminates the common practice of having a separate integration and test phase at the end of the project, which often get compressed or are skipped entirely, resulting in more technical debt and the downward spiral
- Monitoring is important throughout the deployment pipeline
- In addition to monitoring the production service, we also need to monitor the pre-production environments (e.g. dev, test and staging)
- Why? So that we detect and correct potential performance problems long before

production and so that we can also minimize the cost of correcting those problems

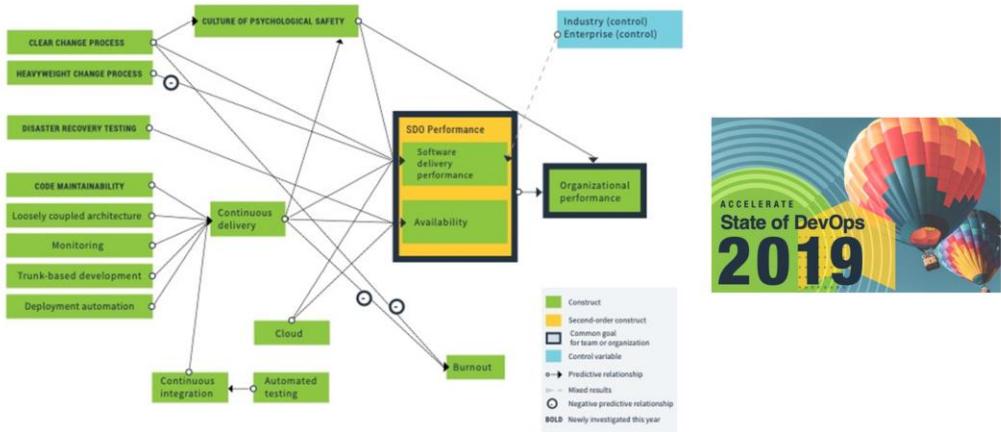
Resources:

- <https://notafactoryanymore.com/2014/08/19/continuous-everything-in-devops-what-is-the-difference-between-ci-cdcd/>



https://youtu.be/xSv_m3KhUO8

Continuous Delivery Leads to Higher Organizational Performance



Module 3: Key DevOps Practices

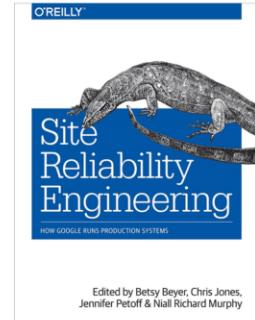
77

Key concepts:

- This graphic is from the 2018 Accelerate State of DevOps report and is a great representation of the ability to increase IT and business performance through DevOps.
- It describes the model the researchers tested in 2018's research.
- It is a structured equation model (SEM), and is a predictive model used to test relationships
- Each box in the figure represents a construct measured in the research, and each arrow represents relationships between the constructs.
- To interpret the model, all arrows can be read using the words *predicts*, *affects*, *drives*, or *impacts*.
- For example, IT performance predicts organizational performance. If you see a (-) next to one of the arrows, it means the relationship is negative: Continuous delivery negatively impacts deployment pain. All arrows in the model represent statistically significant relationships.

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

Module 3: Key DevOps Practices

78

Key Concepts:

- Google created the concept of Site Reliability Engineering and codified it in their book of the same name
- The ideal SRE candidate is a programmer who also has operational, systems or networking knowledge, and likes to whittle down complex tasks
- SRE and DevOps share the same foundational principles. SRE is viewed by many (as cited in the Google SRE book) as a "specific implementation of DevOps with some idiosyncratic extensions." SREs, being developers themselves, will naturally bring solutions that help remove the barriers between development teams and operations teams

Resources:

- <https://landing.google.com/sre/interview/ben-treynor/>

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

Module 3: Key DevOps Practices

79

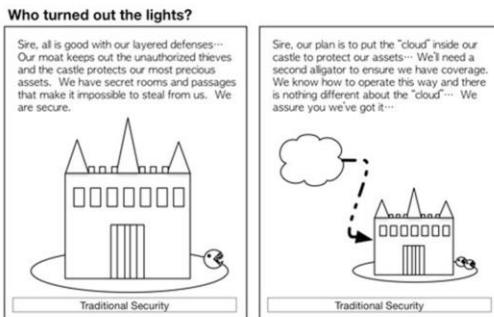
Resources:

- <http://erikhollnagel.com/ideas/resilience-engineering.html>
- <https://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/>
- <https://dzone.com/articles/create-a-culture-of-strength-resilience-engineering>

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



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

Module 3: Key DevOps Practices

80

Key concepts:

- DevSecOps is a principle that embeds security as code by 'shifting left' with continuous security testing throughout the deployment pipeline
- For a while, another concept, Rugged DevOps, was also used in the DevOps security space but DevSecOps has now established itself as the term most popularly used to describe DevOps with a particularly emphasis on security related outcomes. Nowadays, we refer to it as Rugged Software
- Some people are not fans of the term DevSecOps as they believe it could create another silo, demand people learn another thing or still doesn't encapsulate the spirit of what DevOps is really about – however, it's become a fairly well established term and is expected to remain as long as security needs special attention to be brought into the fold
- There are rumored to be over 1 million cybersecurity positions unfilled around the world and some ratios say that for every 100 developers there are 10 IT operations people and 1 security person – DevSecOps breaks the security constraint by spreading the knowledge between humans and embedding it in systems

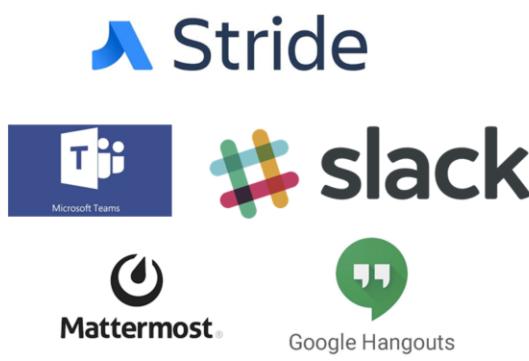
Resources:

- <http://www.devsecops.org/>
- <https://www.sonatype.com/2018survey>

- <https://blog.sonatype.com/start-with-devsecops-reference-architectures>
- <https://www.ruggedsoftware.org/>

ChatOps

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



A screenshot of a Slack interface for a channel named '#deploys'. The sidebar shows 'SpaceCo' with a star icon and a user 'tair'. Below it are sections for 'STARRED' and 'CHANNELS (12)'. The '#deploys' channel is highlighted. A message from 'ahmed' at 10:30 PM says 'so @tair, how about deploying that teleport fe'. A message from 'tair' at 10:32 PM says 'let's do it'. A message from 'Dockbit BOT' at 10:34 PM says 'Deployment of RocketApp by @tair'. A message from '#1014: rocket-app/master' at 10:34 PM says 'Deployment scheduled. Click to see the pro...'. A message from 'tair' at 10:35 PM says 'I'll grab some ☕ while it's deploying...'. A message from 'Dockbit BOT' at 10:35 PM says 'Deployment of RocketApp by @tair'. A message from '#1014: rocket-app/master' at 10:35 PM says '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.

Module 3: Key DevOps Practices

81

Key Concepts:

- ChatOps is a communication approach that allows teams to collaborate and manage many aspects of their infrastructure, code, and data from a chat room
- ChatOps can be a useful tool that decreases MTTR
- ChatOps is great for synchronous communication (regardless of where people are), problem solving and creating a shared history as communications are searchable
- Care must be taken to ensure information that needs
 - To be kept gets moved to a more permanent location such as a knowledgebase, wiki, google doc
 - To be managed longer-term or analyzed gets moved to a process such as incident or problem management

Resources:

- <https://devops.com/chatops-communicating-speed-devops/>
- <https://www.atlassian.com/blog/it-service-management/inside-atlassian-sre-use-chatops-run-incident-management>
- <https://www.infoq.com/news/2018/02/chatops-microsoft-teams/>

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

Key Concepts:

- In Japanese, “Kan” means “visual” and “ban” means “board”
- Kanban is a “pull” system, and so teams pull work only when they are ready for it in an effort to prevent overburden

Resources:

- <https://www.extremeuncertainty.com/misconceptions-about-kanban/>
- <https://www.infoworld.com/article/3266588/a-different-drumbeat-using-kanban-for-devops-to-smooth-out-your-scrum-cycles.html>
- <https://www.upguard.com/blog/devops-kanban-match-heaven>

DISCUSSION

Why Too Much WIP is Bad

Module 3: Key DevOps Practices

Talk about:

- Prioritization
- Context switching
- Completion and finishing work

A note of context switching from Karen Martin's book 'Clarity First':

"To get a sense of the degree to which task switching affects performance and productivity, consider a conservative example: assume that only half of your workforce serve in roles for which task switching and interruptions are the norm; and that the average task switcher loses only 15 minutes of productive time with each switch and that interruptions or other task switching triggers occur only five times a day on average (instead of Meyer's findings of between five and eight times resulting in up to 40 percent loss of productivity).

For a company with 500 full-time employees, that's 250 people experiencing daily productivity losses of 75 minutes each, which add up to nearly one work day per week! Spread out over a year, that productivity loss is equivalent to paying 37 full-time employees to do nothing. Assuming fully loaded labor costs of \$80,000 per employee, that's \$2,960,000 – nearly three million dollars – in labor expense that could be used to absorb growth, invest in innovation, provide greater customer value and improve how the organization operates."

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

Module 3: Quiz Answers

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

Module 3: Key DevOps Practices

Module 4

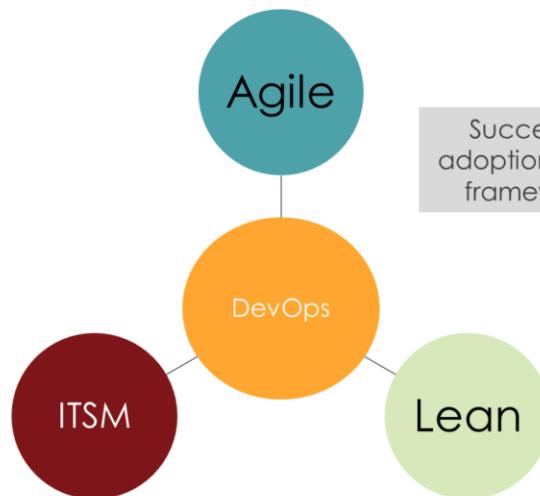
BUSINESS & TECHNOLOGY FRAMEWORKS

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.

Key concepts:

- DevOps isn't a framework or methodology in and of itself
- DevOps adopts and leverages multiple frameworks and methodologies such as agile, lean and IT service management
- Think of DevOps as the convergence of these principles
- DevOps applies Lean principles such as increasing flow and reducing waste to the IT value stream
- DevOps has benefitted tremendously from the work the Agile community has done, showing how small teams, operating with high-trust, small batch sizes with smaller, more frequent software releases, can dramatically increase productivity of Development organizations
- Manufacturing was transformed in the 1980s by moving to Lean production, which enabled organizations that adopted those Lean practices to achieve faster lead times, better quality and to win in the marketplace
- By using those same principles, DevOps enables us to transform how we work in Development and IT Operations
- By doing so, we will not only break the downward spiral but can generate more productivity and economic value for the business

Resource:

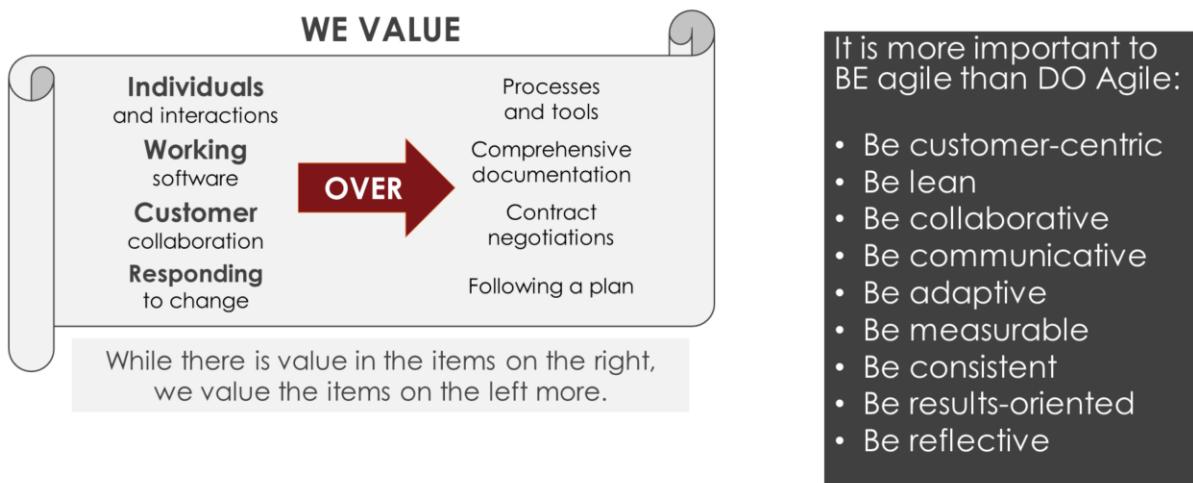
- <http://itrevolution.com/the-convergence-of-devops/>

Agile

Module 4: Business & Technology Frameworks

The Agile Manifesto

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



Key concepts:

- In the late 1990's several methodologies began to get increasing public attention
- Each had a different combination of old ideas, new ideas, and transmuted old ideas, but they all emphasized close collaboration between the programmer team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organizing teams; and ways to craft the code and the team such that the inevitable requirements churn was not a crisis
- The Agile Manifesto was written in February of 2001, at a summit of seventeen independent-minded practitioners of several programming methodologies
- The participants didn't agree about much, but they found consensus around four main values
- Supplementing the Manifesto, the Twelve Principles further explicate what it is to be agile

Resources:

- <http://www.agilemanifesto.org/>

Scrum

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

Scrum.org

Scrum is

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

Scrum increases the ability to release more frequently.

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

Module 4: Business & Technology Frameworks



Roles

- Product Owner
- ScrumMaster
- Development Team



Artifacts

- Product Backlog
- Sprint Backlog
- Increment



Meetings

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

Key concepts:

From the Scrum Guide

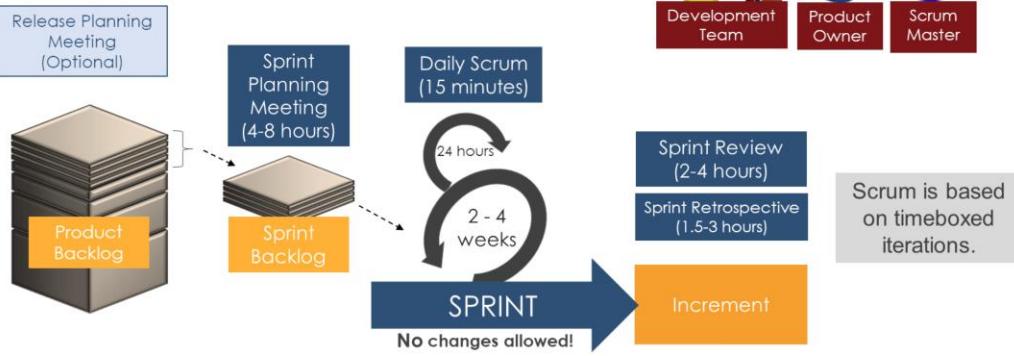
- Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value
- Scrum is:
 - Lightweight
 - Simple to understand
 - Extremely difficult to master
- Scrum is a process framework that has been used to manage complex product development since the early 1990s
- Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques
- Scrum makes clear the relative efficacy of your product management and development practices so that you can improve

Resources:

- <http://www.scrumguides.org/scrum-guide.html>

Scrum in a Nutshell

Scrum = 3 Roles + 3 Artifacts + 5 Events



Module 4: Business & Technology Frameworks

92

Key concepts:

- Product Owner
 - An individual who manages the Product Backlog and ensures the value of the work that the Team performs
- The Team
 - Performs the work and delivers a potentially shippable product
- ScrumMaster
 - An individual who ensures that

the Team adheres to Scrum practices, values and rules

- The Product Owner is responsible for managing the Product Backlog including
 - Clearly expressing Product Backlog items
 - Prioritizing items in the Product Backlog
 - Optimizing the value of the work the Team performs
 - Ensuring that the Product Backlog is visible, transparent and clearly shows what the Team will work on next
 - Ensuring the Team understands items in the Product Backlog
- The Product Owner (one person, not a committee) maintains the Product Backlog and ensures that it is visible to everyone
- Everyone knows what items have the highest priority, so everyone knows what will be worked on
- For the Product Owner to succeed, everyone in the organization has to respect his or her decisions
- No one is allowed to tell the Team to work from a different set of priorities, and the Team isn't allowed to listen to anyone who says otherwise
- The Product Owner needs to understand how to create good user stories or ensure

user stories enable the team to understand what work needs to be done

Resource:

- <http://www.youtube.com/watch?v=XU0lIRlyFM>

The Product Backlog

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



Key Concepts:

- The Product Backlog is where all the requirements are held in the form of User Stories
- User stories can be grouped by Epic as needed
- The Sprint Backlog is the subset of requirements or user stories that will be addressed during the sprint

Resources:

- <https://www.atlassian.com/agile/project-management/epics-stories-themes>

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

94

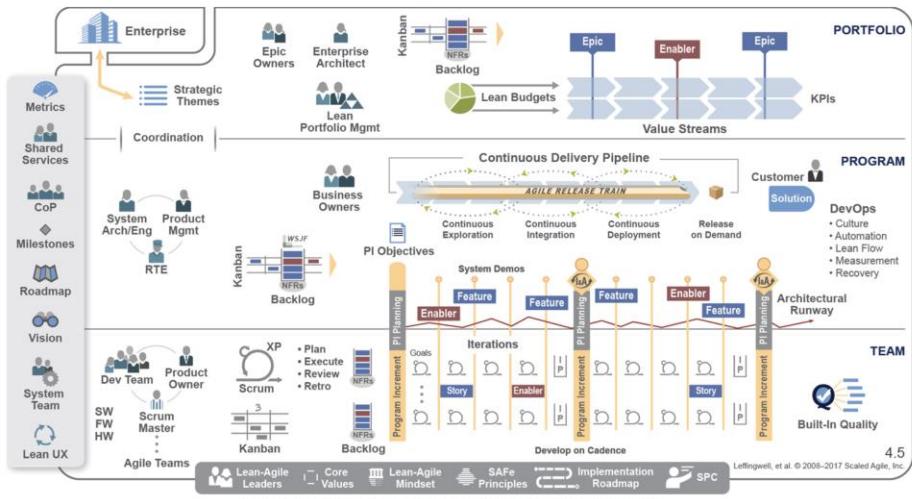
Key concepts:

- SAFe is another agile framework that is gaining popularity
- Learners do not need to have extensive knowledge of SAFe – just a basic understanding of the definition.
- It is in the course to demonstrate that there are multiple approaches to achieve the principles of the Agile Manifesto

Resources:

- <https://auspostenterprise.com.au/insights/digitising-services/australia-posts-agile-approach-digital-transformation>
- <https://www.scaledagileframework.com/safe-lean-agile-principles/>
- <https://www.scaledagileframework.com/case-studies/>

SAFe for Lean Enterprises



95

Key concepts:

In SAFe version 4.5, there are four configurations: essential, portfolio, large solution and full:

- Essential SAFe is the most basic configuration. It describes the most critical elements needed and realizes the majority of the framework's benefits. It includes the team and program level (which it calls agile release trains or ARTs)
- Portfolio SAFe adds the portfolio level to essential SAFe, and allows for such concerns as strategic direction, investment funding, and lean governance
- Large Solution SAFe adds the large solution level to essential SAFe, and allows for coordination and synchronization across multiple programs, but without the portfolio considerations. In earlier versions of SAFe, this level was referred to as value stream
- Full SAFe includes all four levels, building on the team and programs levels in essential SAFe with the large solution and portfolio levels

In the SAFe Spanning Palette:

- Metrics – The primary measure in SAFe is the objective measurement of working solutions. Moreover, SAFe defines some additional intermediate and long-term measures as well, metrics that teams, programs, and portfolios can use to measure progress
- Shared Services – Represents the specialty roles that are necessary for the success

of an ART or value stream, but that cannot be dedicated full time to any specific train

- Community of Practice (CoP) – A community of practice is an informal group of team members and other experts, acting within the context of a program or enterprise, that has a mission of sharing practical knowledge in one or more relevant domains
- Milestones – A milestone is used to track progress toward a specific goal or event. These include fixed-date, Program Increment (PI) and learning milestones
- Roadmap – The roadmap communicates planned ART and value stream deliverables and milestones over a timeline
- Vision – The vision describes a future view of the solution to be developed, reflecting customer and stakeholder needs, as well as Features and Capabilities, which are proposed to address those needs
- System Team – This is a special Agile team that provides assistance in building and using the Agile development environment, including Continuous Integration and test automation and automating the delivery pipeline
- Lean User Experience (UX) – Lean UX is the application of Lean principles to user experience design. It uses an iterative, hypothesis-driven approach to product development, through constant measurement and learning loops (build-measure-learn). In SAFe, Lean UX is applied at scale, with the right combination of centralized and decentralized UX design and implementation

Resource:

- <https://www.scaledagileframework.com/#>

Spotify Engineering Culture Part 1 of 2

Cross-pollination > Standardization: Shows a central 'Spotify X' logo surrounded by three Squads (Squad 1, Squad 2, Squad 3) connected by arrows. Labels include 'Confidence' and 'Flexibility'.

Internal Open-source model: Shows a network of interconnected nodes labeled 'Squad 1', 'Squad 2', and 'Squad 3'.

Alignment: A grid diagram with columns 'low' and 'high' and rows 'low' and 'high'. It includes sketches of people working together and a bridge being built, with labels like 'We need to cross the river', 'Build a bridge', and 'Figure out what's going on'.

Autonomy: Shows a person working alone with a thought bubble: 'I hope someone is working on the same problem'.

Principles > Practices: Shows a hierarchy from 'Scrum' at the top to 'Scrum Master' and 'Agile Coach' below, with a note: 'Scrum > Scrum Master'.

Scrum > Scrum Master: Shows a person thinking: 'Be autonomous, but don't be shortsighted!'

Optional: Shows a person thinking: 'Be a part of something bigger than yourself.'

Community > Structure: Shows a 'Tribe' structure where multiple 'Squads' are grouped together. Labels include 'Community' and 'Structure'.

Focus on Motivation: Shows a person thinking: 'Leaders job: Communicate what problems needs to be solved. And who...'. Includes a survey statistic: '46% everyone. Our employee satisfaction survey says 41% are happy with it, 45% are working here, and 4% are leaving.'

Alignment enables Autonomy: Shows a person thinking: 'Leaders job: Communicate what problems needs to be solved. And who...'. Includes a survey statistic: 'This is of course not satisfactory, and we want to fix it.'

People > *: Shows a person thinking: 'If you're one of those unhappy 4%, please contact us. We're here for your sake, and nothing else.' Includes a graph showing '120+ employees' and '30+ countries'.

Trust > Control: Shows a smiley face turning into a frowny face with the text: '91% → 49%'. Includes a note: 'People > * My colleagues are awesome!'.

Agile at scale requires Trust at scale: Shows a note: 'Politics' and 'Fear'.

YouTube Play Button: A large red YouTube play button icon.

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

Module 4: Business & Technology Frameworks

<https://www.youtube.com/watch?v=4GK1NDTWbkY>

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)

Module 4: Business & Technology Frameworks



Gene Kim

Module 4: Business & Technology Frameworks

99

Resource:

- <https://www.thinktanks.co.za/the-devops-movement-fits-perfectly-with-itsm/>

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.

Key concepts:

- Briefly review each term
- The examinable term on this slide is IT service management (ITSM)
- Every IT organization is executing ITSM processes, whether they have adopted a framework such as ITIL (next slide) or not
- For example, every organization needs to manage changes or fix things (incidents) that break

IT Infrastructure Library® (ITIL®)

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

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

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

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

Key concepts:

- ITIL is the most widely accepted approach to IT service management in the world
- ITIL and ITSM remain the best codifications of the processes that underpin IT Operations, and actually describe many of the capabilities needed in order for IT Operations to support a DevOps-style work stream
- Organizations that have adopted ITIL are finding that they can improve flow by applying agile and lean principles to ITIL process improvement
- Organizations that have adopted DevOps practices but aren't familiar with ITIL find they turn to it for guidance in an effort to improve and streamline their processes
- Once processes have been streamlined, automation can be applied to accommodate the faster lead times and higher deployment frequencies associated with DevOps (particularly relative to Service Transition processes such as Change, Service Asset and Configuration and Release and Deployment Management)
- Start with your key ITIL processes
- ITIL 4 was released in February 2019
- ITIL 4 defines four dimensions that should be considered to ensure a holistic approach to service management
- These dimensions are applicable to the service value system in general and to specific services
- The service value System (SVS) represents "how all the components and activities of an organization work together to facilitate value creation"

Resources:

- <https://devops.com/will-itil-v4-bring-order-to-a-devops-world/>
- <https://www.axelos.com/case-studies-and-white-papers/itil-4-and-devops-a-cultural-perspective>
- <https://www.axelos.com/case-studies-and-white-papers/itil-4-and-digital-transformation>

ITIL 4 Practices: DevOps Touches Them All

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

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

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

Key concepts:

- Discuss the benefits of using models (e.g., faster, more accurate)
- Models are used throughout ITIL (e.g., Incident Management, Problem Management, Release Management)
- Discuss how tools in today's market can be used to help manage and automate processes (workflow)
- Creating models for different pipelines can also help to understand the level of governance, risk and compliance that is needed for each
- Discuss what happens to Project Management in a Product centric world (this is covered in more detail in the DevOps Leader course)

DISCUSSION

Agility in IT Operations

Module 4: Business & Technology Frameworks

- How can we help IT operations people become more agile?
- Can we help IT operations use agile methodologies to manage their work?
- How can we involve IT operations in existing agile processes?
- What would IT operations' relationship with a product backlog look like?
- What events should IT operations participate in and when?
- Who from IT operations would you invite to join in and how? If you are IT operations, when would you want to be involved with what?
- What would be the consequences of IT operations being more agile and having involvement in existing agile processes?

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.

Key Concepts:

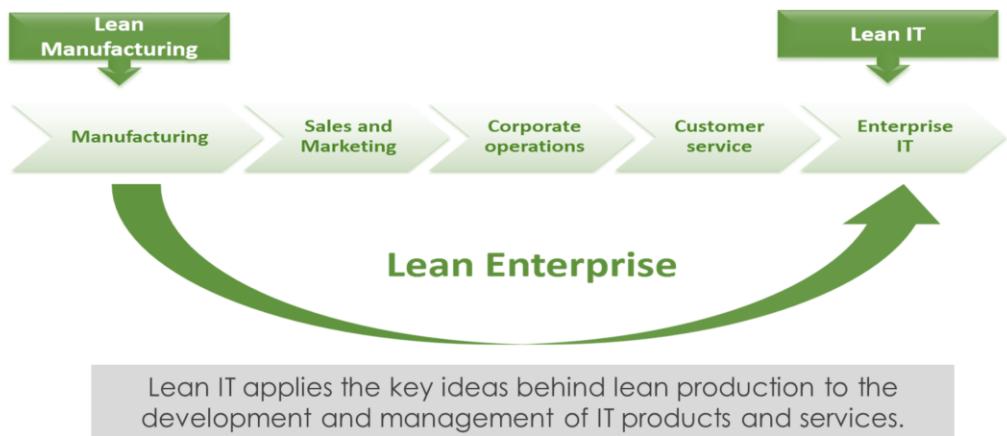
- The development side of many organizations is already becoming comfortable with agile practices
- Agile Service Management bring the vocabulary and discipline of these practices to the service management side of the house; particularly to functions such as Operations Management, the Service Desk, and so forth
- Learning occurs through constant feedback
- Ensures that the values of the Agile Manifesto are embedded in ITSM processes throughout the service lifecycle

Lean

Module 4: Business & Technology Frameworks

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.



Module 4: Business & Technology Frameworks

106

Key concepts:

- Lean's roots are in manufacturing
- Learners may be familiar with the term 'Lean Manufacturing'
- Simply put, Lean Manufacturing is the application of lean production principles to the manufacturing unit of an organization... in the same way that Lean IT is the application of lean production principles to the IT organization
- Lean Enterprise would be a strategic initiative, typically driven from the top
- A successful Lean Enterprise initiative would require empowering everyone in the organization to participate in Lean

Resource:

- <http://www.leanproduction.com/top-25-lean-tools.html>

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

Key concepts:

- Toyota executive Taiichiro Ohno first identified 7 sources of waste as part of the Toyota Production System (TIMWOOD)
- More recently, others have added to and modernized the sources of waste from TIMWOOD into the acronym 8 waste acronym DOWNTIME. This list is examinable
- There may be other sources of waste. There is an exercise for “where’s the waste” in the exercise deck.
- Review these sources of waste and provide examples in preparation for the next exercise
- Examples may include:
 - Unnecessary code or functionality
 - Starting more than can be completed
 - Delays in the software development process
 - Unclear or constantly changing requirements
 - Bureaucracy
 - Slow or ineffective communication
 - Partially done work
 - Defects and quality issues
 - Task switching

EXERCISE

Identifying & Eradicating Waste

Module 4: Business & Technology Frameworks

Using the previous slide, identify a key area of waste in your organization and then come up with 3 ways in which you could work to eradicate it.



Gene Kim

Module 4: Business & Technology Frameworks

109

“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.”

Key concepts:

- In 2010 at a DevOps Days in Mountainview, California, John Willis and Damon Edwards coined the DevOps acronym CAMS, which stands for Culture, Automation, Measurement and Sharing.
- Jez Humble, author of the Continuous Delivery book later added an L, standing for Lean, to form CALMS.
- Since then Lean principles and tools have been considered to be a core competency and value for DevOps as reflected in Beyond The Phoenix Project, an audiobook produced by John Willis and Gene Kim, that devotes a module to the subject.

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.”

“Alaska views itself as a tech company with wings.”



Veresh Sita, CIO



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

- <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/mob-programming-at-alaska-airlines-with-agile-amped-at-aatc2016/>

Value Stream Mapping

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

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



Module 4: Business & Technology Frameworks

111

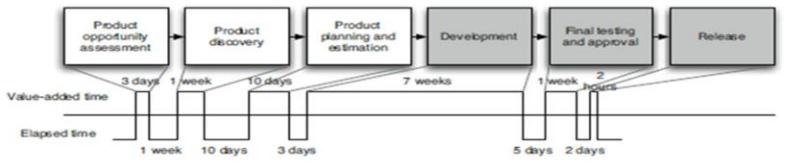
Key Concepts:

- Value stream mapping helps organizations analyze the current state and design a future state for the activities that take a product or service from its beginning through to the customer
- A value stream map is used to understand and streamline work processes using Lean tools and techniques
- A key is to see things from a customer perspective and strive to reduce waste, vs. value-adding activities
- Effective value Stream Mapping involves observing the process directly (vs. visualizing the value stream in an office or conference room)
- A value stream map makes work visible to everyone
- It's a dynamic document that gets refined as processes are improved

Resources:

- <https://www.infoq.com/articles/waste-not-want-not/>
- <https://www.industryweek.com/continuous-improvement/best-practices-using-value-stream-mapping-continuous-improvement-tool>
- <https://youtu.be/gg5u9kn0Bzo>

Sample Value Stream Maps



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



Source: Ranger4

Module 4: Business & Technology Frameworks

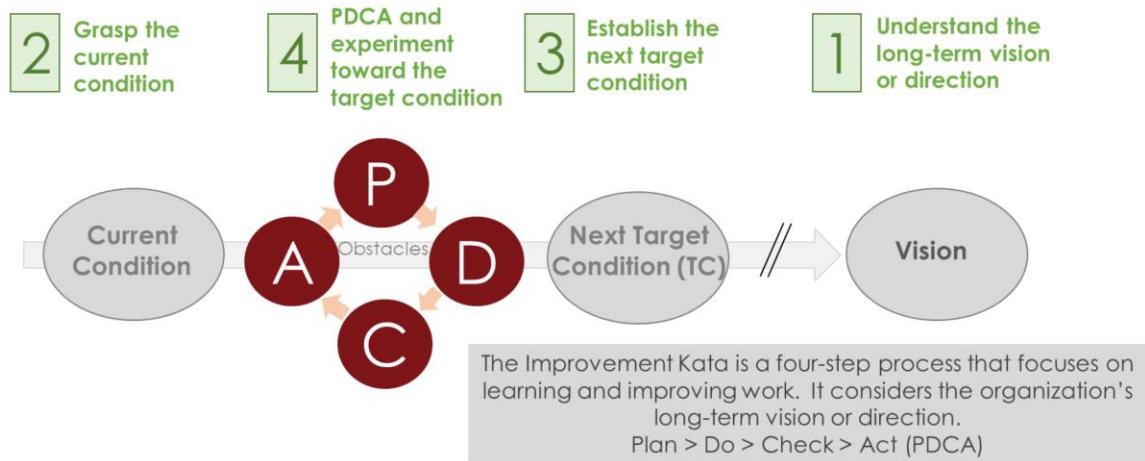
112

Key concepts:

These are two visualizations of the Value Stream, which are often built in organizations with sticky notes that show how and where a process or unit contributes value or represents waste.

Improvement Kata

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



Module 4: Business & Technology Frameworks

Key concepts:

- The Improvement Kata is a structured and focused approach to create a continuous learning and improvement culture – a kaizen culture (Good change)
- To make the scientific Improvement Kata pattern a habit in an organization its managers teach/coach the Improvement Kata routines a little bit every day
- In contrast to continuous improvement approaches that attempt to predict the path and focus on implementation, the improvement kata builds on discovery that occurs along the way
- **Teams using the Improvement Kata learn as they strive to reach a target condition, and adapt based on what they are learning**
- The five questions are used by a coach to guide a learner through the PDCA cycles required to overcome obstacles standing in the way of the target condition
- Martial arts practitioners will be familiar with the term ‘kata’ as the repetition of a move that makes it natural and an instinctive habit

Resource:

- <http://www.methodsandtools.com/archive/toyotakata.php>

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

Key concepts:

- Safety culture in DevOps is about both
 - Psychological safety – rewarding the right behaviors and being fearless about reporting problems
 - Systemic safety – using techniques and tooling to prevent, preempt and predict and remediate failure
- Safety Culture, as a term, emerged after the Chernobyl nuclear disaster and has been documented extensively by Sidney Dekker
 - The airlines that report the most incidents are statistically proven to be the safest
- The Andon Cord is a cord in the Toyota factories that caused the line to stop when anyone spots a defect. It drives a number of behaviors:
 - A ‘come see’ moment
 - Thanks for the learning opportunity
 - The prioritization of the problem

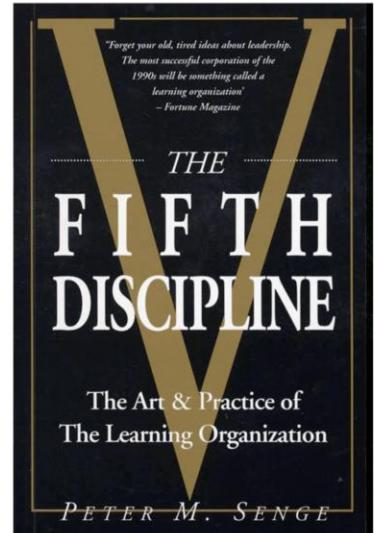
Resources:

- Sidney Dekker, Richard Cook and Stephen Spear at DOES 2017 -
<https://www.youtube.com/watch?v=CFMJ3V4VakA>
- <https://www.infoq.com/news/2018/05/Beyond-The-Phoenix-Project>

- The Field Guide to Understanding Human Error:
<https://static1.squarespace.com/static/53b78765e4b0949940758017/t/581c73ac29687fefa2d05bd2/1480886082793/Field+Guide+to+Understanding+Human+Error+Sidney-Dekker.pdf>
- Just Culture: <http://sidneydekker.com/just-culture/>
- <https://itrevolution.com/kata/> - the andon cord

Learning Organizations

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



Module 4: Business & Technology Frameworks

115

Key Concepts:

- Learning organizations understand that not embedding learning into the culture of an organization create cultural debt
- Organizations that don't learn are less likely to be able to compete long term and may experience high (and expensive) staff turnover
- At the end of 'Beyond the Phoenix Project', John Willis and Gene Kim conclude that they will refer to high performing organizations that practice DevOps principles as 'Dynamic Learning Organizations'

Resources:

- <https://hbr.org/1993/07/building-a-learning-organization>
- <https://hbr.org/2008/03/is-yours-a-learning-organization>

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



Key Concepts:

- Annual budgeting cycles are not agile enough for agile
- At DOES 2016, Jon Smart from Barclays said that they had gone 4% to 50% of strategic change projects being agile in the last 12 months – Helen Beal asked if he was continually funding them – the Twitter screenshot on this slide shows his response – a step towards a continuous funding model

Resources:

- Beyond Budgeting: <https://bbrt.org/>
- Let's Fund Teams Not Projects (DEFRA – UK Government):
<https://defradigital.blog.gov.uk/2017/09/19/lets-fund-teams-not-projects/>
- <https://devops.com/transforming-the-annual-budgeting-process-for-devops/>

Module 4: Quiz

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

Module 4: Quiz Answers

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

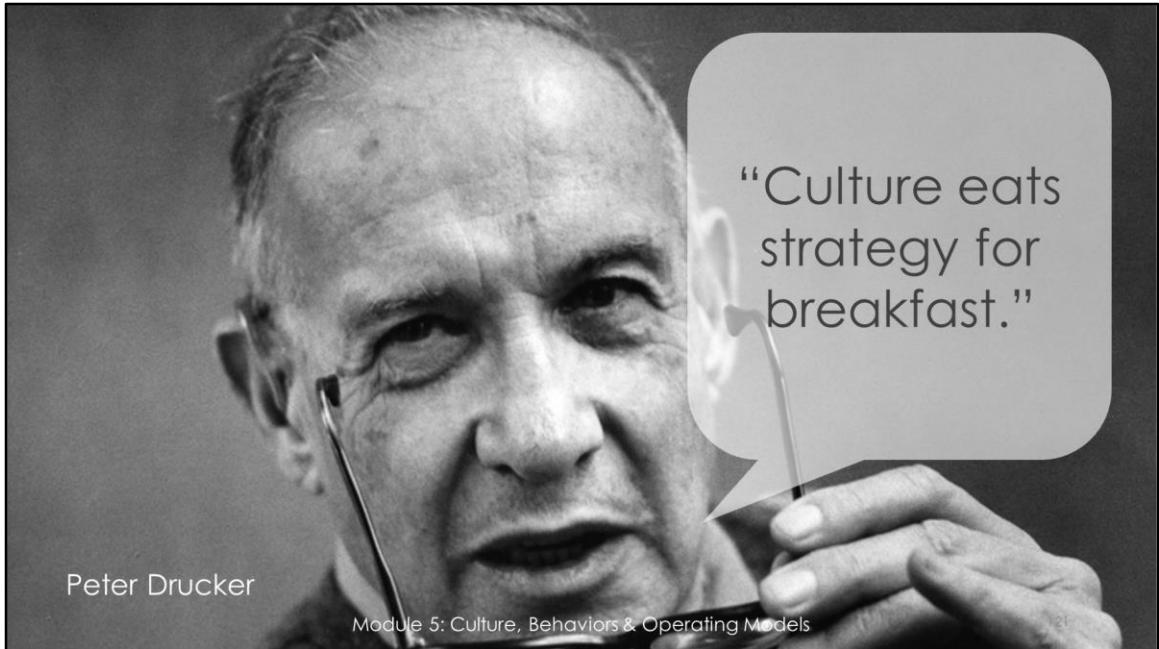
Module 5

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



Peter Drucker

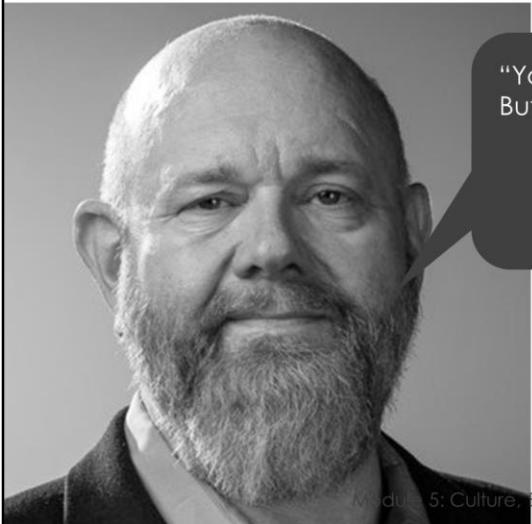
Module 5: Culture, Behaviors & Operating Models

21

Key concepts:

- Peter Drucker was a management consultant whose writings contributed to the philosophical and practical foundations of the modern business corporation
- Peter Drucker invented the concept known as management by objective
- In 2006, Mark Fields of Ford Motor Company attributed the quote on this slide to Peter Drucker
- Peter Drucker often argued that a company's culture would trump any attempt to create a strategy that was incompatible with its culture

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

Key concepts:

- Culture is quite a nebulous term and difficult to define and quantify
- Focusing on behaviors and values helps to address these difficulties as we can understand how an organization of a number of individuals deals with things like blame, fear, failure, learning, innovation, experimentation, trust, respect etc

DevOps Helps to Overcome Cultural Debt

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



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

Dharmesh Shah, Founder & CTO



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

Module 5: Culture, Behaviors & Operating Models

123

Key concepts:

Cultural debt happens when:

- Silos become impenetrable (fiefdoms)
- Companies hire the wrong people
- Employees don't feel empowered
- Employees don't feel their contributions matter
- Peoples' contributions aren't acknowledged
- People aren't given the time or resources needed to make improvements
- Feedback loops are negative or non-existent
- Information is hidden
- People put what's happening today over investing in the future

Resources:

- <http://devops.com/blogs/culture-debt/>
- <http://www.bigvisible.com/2014/04/becoming-agile-so-how-long-does-this-take/>
- https://www.slideshare.net/dharmeshs/culture-code-creating-a-company-you-love/21-Cultural_debt_is_rarelyforgivenYoure_going

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.

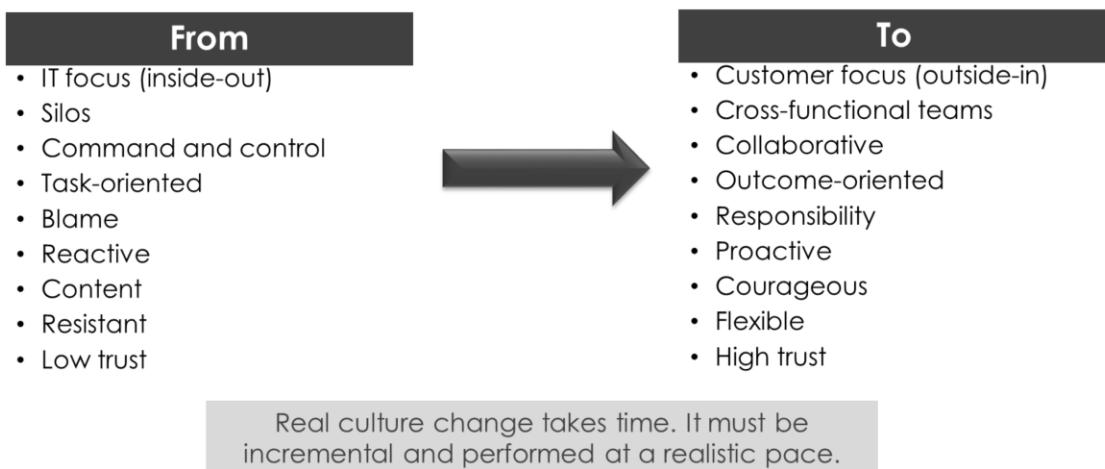
Key Concepts:

- Having a shared vision, goals and incentives is critical – this is what will break down silos
- An example of a ‘safe’ environment is one where people feel comfortable speaking up (e.g., blameless postmortems)
- A safe environment requires high trust and a learning culture
- Some companies recognize teams or individuals that enable the organization to learn from failure
- Such awards send a message about what company values

Resource:

- <https://www.wired.com/insights/2013/06/building-a-healthy-devops-culture/>
- <https://www.catalign.in/2010/12/4-characteristics-of-innovative-culture.html>
- <http://commsmasters.com/2012/10/is-your-organisations-culture-killing-open-honest-communication/>

Shifting Thoughts and Behaviors



Module 5: Culture, Behaviors & Operating Models

125

Key concepts:

- Organizations may find they need to tackle one behavior at a time as many are interrelated
- Look at people's current behavior to determine a starting point

Resource:

<http://devops.com/blogs/evolve-devops/>

High Trust vs. Low Trust



Key concepts:

- High-trust organizations encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas
- They also empower people which enables people to move more quickly as they don't have to wait for someone else to make a decision or take action
- As an example, organizations that require a high percentage of changes to be approved by a Change Advisory Board that only meets weekly is going to slow down the Change Management process and increase the cost of handling changes (with, according to **the 2014 State of DevOps Report**, no impact on restore times and only a negligible effect on reducing failed changes)
- Conversely, organizations that increase their use of standard changes can make lower risk changes more quickly at a lower cost

Resources:

- <http://www.oeeea.com/2013/01/8-signs-that-you-are-working-in-low.html>

Spotify Engineering Culture
Part 2 of 2
(Henrik Kniberg Apr 2016)

Continuous Improvement
Driven from below. Separated from above.

Fail Fast → Learn Fast → Improve fast

Fail-friendly environment

Failing Reverses → Failure Ambiance

Limited Blast Radius
via Devoted Architecture
via Gradual Rollout

Experiment-friendly Culture

A or B?
try both and compare

Data-driven decisions

Waste-repellent Culture via Lean
If it works, keep it. Otherwise, change it.

Keep
• Retrospectives
• Daily Standup
• Google Docs
• GitHub
• Small User Stories

Skip/dump
• Time reports
• Deadlines
• Large task teams
• Test phases
• Task schedules
• Standup meetings
• Corporate BS

Chaos > Bureaucracy
Agile
Cross
Weekly Demo

Impact > Velocity
Banking, Ditching, Refactoring, Optimized

Impact vs. Test
Impact control

Idea/Problem
Narrative + Pictures

Build MVP
Tweak → Release

Analyze Data

Innovation > Predictability
100% predictability = 0% innovation

Hack Time
People are natural innovators.
= 10%

Spotify Hack Week
Do whatever! Where whenever!
In whatever way!

Minimize the need for Big Projects
Big Project = Big Risk

Visual Progress
Cross Wall

Module 5: Culture, Behaviors & Operating Models



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

<https://www.youtube.com/watch?v=rzoyryY2STQ>

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.

Key concepts:

- This study was initially conducted to look how culture affects the performance of a medical unit, particularly as it relates to safety
- The concepts are very applicable, however, to any type of organization
- Westrum's study looked at how organizations respond to problems and opportunities
- The types described – pathological, bureaucratic, and generative – are shaped by the preoccupations of the organization's leaders
- In other words, team leaders shape the organization's culture by creating incentive structures that reward certain behaviors
- In the context of DevOps, techniques we can use to create and maintain a high-trust culture include
 - Encouraging and creating boundary-spanning teams
 - Making quality, availability and security everyone's responsibility, instead of just Ops
 - Holding blameless post-mortems when incidents and outages occur to develop effective countermeasures and create global learning
 - Maximizing everyone's creativity to find novel solutions to problems
- You don't need to memorize the three types of organizations (pathological, etc.)

- You should, however, be able to recognize the characteristics of a high-trust (generative) culture
- According to the 2015 State of DevOps Report, burnout is associated with pathological cultures
- Bureaucratic cultures are likely to experience the morale issues that come from having narrow responsibilities or that come from an ‘us’ vs. ‘them’ culture
- Pay close attention to the red text for high-trust (generative) organizations

Additional resource:

- <https://cloud.google.com/solutions/devops/devops-culture-westrum-organizational-culture>

EXERCISE

Rating & Improving Using the Westrum Model

Using the previous slide, the Westrum Typology of Organizational Culture, choose one row, rate your organization as Pathological, Bureaucratic or Generative and, if you do not rate yourselves as Generative, come up with some ideas of changes you could make to improve your capability rating.



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

Peter Senge

Culture Change is Never Easy

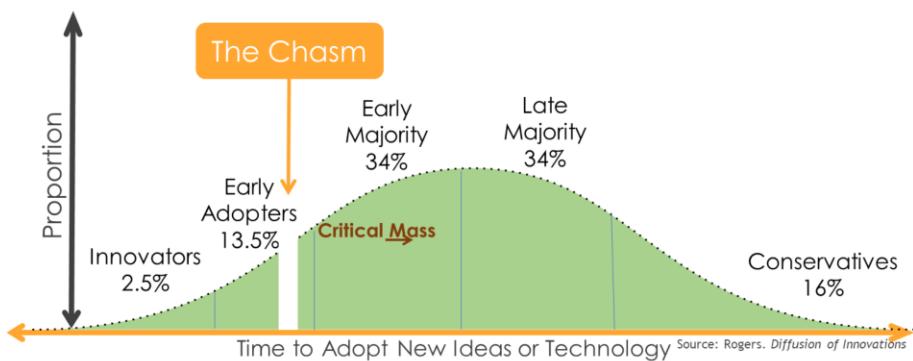
People typically don't resist their own ideas.

- 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

Key concepts:

- To adopt DevOps practices, the culture of the company must enable and encourage collaboration
- The culture has to make it okay for people to make mistakes and speak their minds
- Organizational change management practices must be introduced
- Organizational change management provides the preparation, motivation and education people need to embrace and support change

People Adapt to Change at Different Paces



Adoption means that a person does something differently than before.

Key Concepts:

- Innovators get early adopters on board, then Innovators work with Early Adopters to get the Early Majority's buy in, then the three groups "gang up" on the late majority and win them over
- You don't worry about the Conservatives because they will either join or leave
- Diffusion of Innovation (DOI) Theory, developed by E.M. Rogers in 1962, is one of the oldest social science theories. It originated in communication to explain how, over time, an idea or product gains momentum and diffuses (or spreads) through a specific population or social system. The end result of this diffusion is that people, as part of a social system, adopt a new idea, behavior, or product. Adoption means that a person does something differently than what they had previously (i.e., purchase or use a new product, acquire and perform a new behavior, etc.). The key to adoption is that the person must perceive the idea, behavior, or product as new or innovative. It is through this that diffusion is possible.
- Adoption of a new idea, behavior, or product (i.e., "innovation") does not happen simultaneously in a social system; rather it is a process whereby some people are more apt to adopt the innovation than others. Researchers have found that people who adopt an innovation early have different characteristics than people who adopt an innovation later. When promoting an innovation to a target

population, it is important to understand the characteristics of the target population that will help or hinder adoption of the innovation.

- There are **five established adopter categories**, and while the majority of the general population tends to fall in the middle categories, it is still necessary to understand the characteristics of the target population. When promoting an innovation, there are different strategies used to appeal to the different adopter categories.
 1. **Innovators** – These are people who want to be the first to try the innovation. They are venturesome and interested in new ideas. These people are very willing to take risks, and are often the first to develop new ideas. Very little, if anything, needs to be done to appeal to this population.
 2. **Early Adopters** – These are people who represent opinion leaders. They enjoy leadership roles, and embrace change opportunities. They are already aware of the need to change and so are very comfortable adopting new ideas. Strategies to appeal to this population include how-to manuals and information sheets on implementation. They do not need information to convince them to change.
 3. **Early Majority** – These people are rarely leaders, but they do adopt new ideas before the average person. That said, they typically need to see evidence that the innovation works before they are willing to adopt it. Strategies to appeal to this population include success stories and evidence of the innovation's effectiveness.
 4. **Late Majority** – These people are skeptical of change, and will only adopt an innovation after it has been tried by the majority. Strategies to appeal to this population include information on how many other people have tried the innovation and have adopted it successfully.
 5. **Conservatives (Laggards)** - These people are bound by tradition and very conservative. They are very skeptical of change and are the hardest group to bring on board. Strategies to appeal to this population include statistics, fear appeals, and pressure from people in the other adopter groups.

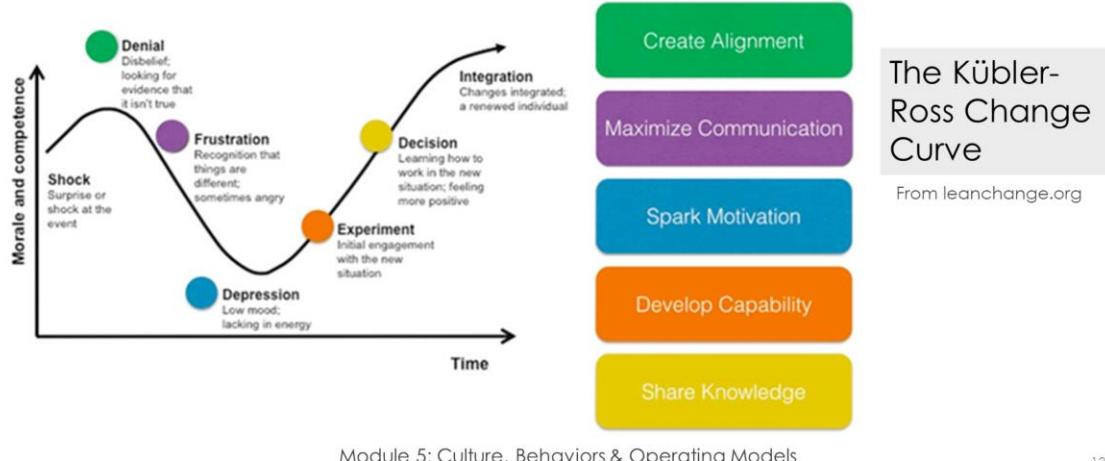
DISCUSSION

Placing on the Change Curve

Think about the previous slide and discuss where:

- You think DevOps is overall in terms of market adoption
- Where your organization is in terms of DevOps adoption
- Where you would place yourself as an individual within your organization in terms of your own DevOps adoption

The Stages of Change Acceptance



Key concepts:

- Where are you personally on this curve?
- Where is your organization?
- Are there ways to reduce the impact of change?

Resources:

- Elisabeth Kübler-Ross was a Swiss-American psychiatrist who introduced this model to reflect stages of grief that can occur in any order at any time
- The model also holds true when it comes to business, work or employment
- <http://www.cleverism.com/understanding-kubler-ross-change-curve/>

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

Module 5: Culture, Behaviors & Operating Models

Key concepts:

- Chat platforms include HipChat, Stride, FlowDock, Slack, Microsoft Teams/Lync/Skype/OCS, WhatsApp
- Task managers include Trello, Asana, Basecamp, Jira, VSTS
- Social tools include Facebook, Twitter, LinkedIn, blogs, Yammer, Jive
- Alert management tools include PagerDuty and Raygun
- Knowledge sharing platforms include wikis, Github , Confluence

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?

Key concepts:

- Collaboration enables people to work together to achieve goals that they could not reach individually
- Collaboration methods should be evaluated periodically to determine if they are achieving the desired result and to ensure parity
- Communication is passive – I tell you, you listen to me (and vice versa)
- Collaboration is active – I ask for your input, feedback and opinion (what do *you* think?)
- Can you force people to collaborate? Set up meetings? Arrange workspaces? Demand reports? What kind of leadership can support collaboration and how?

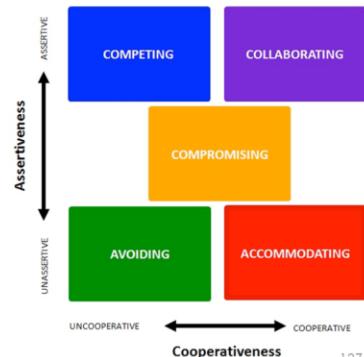
Expect Some Conflict: Thomas-Kilmann Conflict Modes

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



Module 5: Culture, Behaviors & Operating Models

137

Key concepts:

- Many individuals avoid conflict of any kind, particularly when it appears to risk the success of the project
- Organization Change Management (OCM) actually improves when conflict is acknowledged and opposing or conflicting opinions allowed the chance to be heard
- Everyone is capable of using each of these modes – however, we most likely have a “default” mode that may be overused or used in the wrong situation.
- Understanding other people’s use of these modes can help leaders adapt their own behavior and potentially that of others around them
- When are which modes appropriate and inappropriate?
- Competing** is assertive and uncooperative—an individual pursues his own concerns at the other person's expense. This is a power-oriented mode in which you use whatever power seems appropriate to win your own position—your ability to argue, your rank, or economic sanctions. Competing means "standing up for your rights," defending a position which you believe is correct, or simply trying to win.
- Accommodating** is unassertive and cooperative—the complete opposite of competing. When accommodating, the individual neglects his own concerns to

satisfy the concerns of the other person; there is an element of self-sacrifice in this mode. Accommodating might take the form of selfless generosity or charity, obeying another person's order when you would prefer not to, or yielding to another's point of view.

- **Avoiding** is unassertive and uncooperative—the person neither pursues his own concerns nor those of the other individual. Thus he does not deal with the conflict. Avoiding might take the form of diplomatically sidestepping an issue, postponing an issue until a better time, or simply withdrawing from a threatening situation.
- **Collaborating** is both assertive and cooperative—the complete opposite of avoiding. Collaborating involves an attempt to work with others to find some solution that fully satisfies their concerns. It means digging into an issue to pinpoint the underlying needs and wants of the two individuals. Collaborating between two persons might take the form of exploring a disagreement to learn from each other's insights or trying to find a creative solution to an interpersonal problem.
- **Compromising** is moderate in both assertiveness and cooperativeness. The objective is to find some expedient, mutually acceptable solution that partially satisfies both parties. It falls intermediate between competing and accommodating. Compromising gives up more than competing but less than avoiding. Likewise, it addresses an issue more directly than avoiding, but does not explore it in as much depth as collaborating. In some situations, compromising might mean splitting the difference between the two positions, exchanging concessions, or seeking a quick middle-ground solution.

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.

Key concepts:

- Change fatigue can sever peoples' ability to be committed to the organization
- When too much is changing at one time – especially when that change is dictated – people can start to feel helplessness
- People can begin to perceive that things are never going to get better and there is nothing they can do about it
- People also sometimes need help connecting the dots
- Show how changes are connected
- More importantly show people that what they are doing is contributing to the change in a positive way
- People are sometimes working on a project such as an automation project and don't necessarily understand that it is part of the greater DevOps initiative

Resources:

- <http://humanresources.about.com/od/resistancetochange/a/how-to-reduce-resistance-to-change.htm>

Empower New Behaviors

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

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

Key Concepts:

- Many of these activities are aimed at engaging and educating existing employees
- According to a research report 'What Smart Businesses Know About DevOps,' over 70% of DevOps leaders are retraining existing personnel, while only 50% are hiring new resources
- Communities of practice can play in enable people to share their knowledge

Resources:

- <https://techbeacon.com/devops/7-devops-roles-you-need-succeed>
- <https://www.infoq.com/articles/communities-of-practice-agile-organisation/>

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

<https://www.youtube.com/watch?v=7s-VbB1fG5o&feature=youtu.be>

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

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 |

Module 6

AUTOMATION & ARCHITECTING DEVOPS TOOLCHAINS

Module 6: Automation & Architecting DevOps Toolchains

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

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



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

Christopher Little, quoted in The DevOps Handbook

Key concepts:

- DevOps is not just about automation but there are common enabling practices found in organizations that are adopting a DevOps culture
- DevOps extends and builds upon the practices of ‘infrastructure as code’ pioneered by Dr. Mark Burgess
- Infrastructure as code enables the reconstruction of the business from nothing but a source code repository, an application data backup, and bare metal resources (Jessie Robbins)
- Mention that we’ll discuss continuous integration, continuous delivery and continuous deployment in greater detail later in the course
- On-demand creation of environments and keeping those environment in sync is critical
- Every environment is *someone’s* production environment as it is ultimately where they do their work

Resources:

- <http://www.thoughtworks.com/insights/blog/jesse-robbins-discusses-devops-and-cloud-computing>

PERIODIC TABLE OF DEVOPS TOOLS (V3)

The Periodic Table of DevOps Tools (V3) is a grid where each element represents a different tool or technology. The elements are color-coded based on their primary function:

- Open Source**: Os (Purple)
- Free**: Fr (Orange)
- Freemium**: Fm (Blue)
- Paid**: Pd (Red)
- Enterprise**: En (Dark Blue)

Legend:

- Source Control Mgmt.**: Purple
- Deployment**: Green
- Analytics**: Teal
- Monitoring**: Dark Gray
- Database Automation**: Orange
- Continuous Integration**: Light Blue
- Release Orchestration**: Green
- Security**: Blue
- Cloud**: Yellow
- Collaboration**: Magenta
- Configuration**: Dark Blue
- AL/Ops**: Brown

The table includes the following elements:

Period	Group	Element	Description
1	Os	Gl	GitLab
2	Fr	Gh	Github
3	Fm	Dt	Octopus
4	Pd	Sv	Subversion
5	En	Db	Drakonator
6	Os	Cw	ISFW
7	Fr	Dp	Dashplate
8	Fm	Jn	Jenkins
9	Pd	Cs	CodeShip
10	En	Fn	Filtesse
11	Os	Ju	JUnit
12	Fr	Ka	Karma
13	Fm	Su	ScrumUp
14	Pd	Ch	Chef
15	En	Tf	Terraform
16	Fr	XLd	RobotLife Deployer
17	Fm	Ur	UrbanCode Release
18	Pd	Ld	Lambda
19	En	Af	Azure Functions
20	Os	Ic	Amazon CloudWatch Metrics
21	Fr	Fd	FaaS
22	Pd	Sg	Sumo Logic
23	En	Sp	Spank
24	Os	XLr	XLabs Lambda SL Release
25	Fr	Aws	AWS Lambda
26	Fm	Az	Azure Functions
27	Pd	Gc	Google Cloud Functions
28	En	Op	OpenShift
29	Os	Sg	Sumo Logic
30	Fr	Dk	Docker
31	Pd	Ld	Lambda
32	En	Ic	Amazon CloudWatch Metrics
33	Os	Fd	FaaS
34	Fr	Sg	Sumo Logic
35	Pd	Sp	Spank
36	En	Ps	Prometheus
37	Os	Pr	Plutora Release
38	Fr	Al	Amazon Lambda
39	Pd	Os	OpenTracing
40	En	Ps	Prometheus
41	Os	Gke	GKE
42	Fr	Om	OpenMake
43	Pd	Cp	Apache Continuous Delivery Pipeline
44	En	Cy	ITMS
45	Os	GkCD	GKE Continuous Delivery
46	Fr	Ra	ElectricCloud Rancher
47	Pd	Rk	Rkt
48	En	Sp	Spinnaker
49	Os	Aks	AKS
50	Fr	It	ITMS
51	Pd	Ns	Nomad
52	En	Mg	ManageIQ
53	Os	Ir	Iron.io
54	Fr	Ls	Logstash
55	Pd	Aw	Apache OpenWhisk
56	En	Bd	BlackDuck
57	Os	Sr	SonarQube
58	Fr	Hv	HiveMind
59	Pd	Hv	HiveMind
60	En	Wp	Web Services API
61	Os	Wp	Web Services API
62	Fr	Cm	Cloudbees AST
63	Pd	Bd	BlackDuck
64	En	Sr	SonarQube
65	Os	Hv	HiveMind
66	Fr	Hv	HiveMind
67	Pd	Hv	HiveMind
68	En	Wp	Web Services API
69	Os	Cm	Cloudbees AST
70	Fr	Bd	BlackDuck
71	Pd	Sr	SonarQube
72	En	Hv	HiveMind
73	Os	Hv	HiveMind
74	Fr	Wp	Web Services API
75	Pd	Cm	Cloudbees AST
76	En	Bd	BlackDuck
77	Os	Sr	SonarQube
78	Fr	Hv	HiveMind
79	Pd	Hv	HiveMind
80	En	Wp	Web Services API
81	Os	Cm	Cloudbees AST
82	Fr	Bd	BlackDuck
83	Pd	Sr	SonarQube
84	En	Hv	HiveMind
85	Os	Wp	Web Services API
86	Fr	Cm	Cloudbees AST
87	Pd	Bd	BlackDuck
88	En	Sr	SonarQube
89	Os	Hv	HiveMind
90	Fr	Wp	Web Services API
91	Pd	Cm	Cloudbees AST
92	En	Bd	BlackDuck
93	Os	Sr	SonarQube
94	Fr	Hv	HiveMind
95	Pd	Wp	Web Services API
96	En	Cm	Cloudbees AST
97	Os	Bd	BlackDuck
98	Fr	Sr	SonarQube
99	Pd	Hv	HiveMind
100	En	Wp	Web Services API
101	Os	Cm	Cloudbees AST
102	Fr	Bd	BlackDuck
103	Pd	Sr	SonarQube
104	En	Hv	HiveMind
105	Os	Wp	Web Services API
106	Fr	Cm	Cloudbees AST
107	Pd	Bd	BlackDuck
108	En	Sr	SonarQube
109	Os	Hv	HiveMind
110	Fr	Wp	Web Services API
111	Pd	Cm	Cloudbees AST
112	En	Bd	BlackDuck
113	Os	Sr	SonarQube
114	Fr	Hv	HiveMind
115	Pd	Wp	Web Services API
116	En	Cm	Cloudbees AST
117	Os	Bd	BlackDuck
118	Fr	Sr	SonarQube
119	Pd	Hv	HiveMind
120	En	Wp	Web Services API
121	Os	Cm	Cloudbees AST
122	Fr	Bd	BlackDuck
123	Pd	Sr	SonarQube
124	En	Hv	HiveMind
125	Os	Wp	Web Services API
126	Fr	Cm	Cloudbees AST
127	Pd	Bd	BlackDuck
128	En	Sr	SonarQube
129	Os	Hv	HiveMind
130	Fr	Wp	Web Services API
131	Pd	Cm	Cloudbees AST
132	En	Bd	BlackDuck
133	Os	Sr	SonarQube
134	Fr	Hv	HiveMind
135	Pd	Wp	Web Services API
136	En	Cm	Cloudbees AST
137	Os	Bd	BlackDuck
138	Fr	Sr	SonarQube
139	Pd	Hv	HiveMind
140	En	Wp	Web Services API
141	Os	Cm	Cloudbees AST
142	Fr	Bd	BlackDuck
143	Pd	Sr	SonarQube
144	En	Hv	HiveMind
145	Os	Wp	Web Services API
146	Fr	Cm	Cloudbees AST
147	Pd	Bd	BlackDuck
148	En	Sr	SonarQube
149	Os	Hv	HiveMind
150	Fr	Wp	Web Services API
151	Pd	Cm	Cloudbees AST
152	En	Bd	BlackDuck
153	Os	Sr	SonarQube
154	Fr	Hv	HiveMind
155	Pd	Wp	Web Services API
156	En	Cm	Cloudbees AST
157	Os	Bd	BlackDuck
158	Fr	Sr	SonarQube
159	Pd	Hv	HiveMind
160	En	Wp	Web Services API
161	Os	Cm	Cloudbees AST
162	Fr	Bd	BlackDuck
163	Pd	Sr	SonarQube
164	En	Hv	HiveMind
165	Os	Wp	Web Services API
166	Fr	Cm	Cloudbees AST
167	Pd	Bd	BlackDuck
168	En	Sr	SonarQube
169	Os	Hv	HiveMind
170	Fr	Wp	Web Services API
171	Pd	Cm	Cloudbees AST
172	En	Bd	BlackDuck
173	Os	Sr	SonarQube
174	Fr	Hv	HiveMind
175	Pd	Wp	Web Services API
176	En	Cm	Cloudbees AST
177	Os	Bd	BlackDuck
178	Fr	Sr	SonarQube
179	Pd	Hv	HiveMind
180	En	Wp	Web Services API
181	Os	Cm	Cloudbees AST
182	Fr	Bd	BlackDuck
183	Pd	Sr	SonarQube
184	En	Hv	HiveMind
185	Os	Wp	Web Services API
186	Fr	Cm	Cloudbees AST
187	Pd	Bd	BlackDuck
188	En	Sr	SonarQube
189	Os	Hv	HiveMind
190	Fr	Wp	Web Services API
191	Pd	Cm	Cloudbees AST
192	En	Bd	BlackDuck
193	Os	Sr	SonarQube
194	Fr	Hv	HiveMind
195	Pd	Wp	Web Services API
196	En	Cm	Cloudbees AST
197	Os	Bd	BlackDuck
198	Fr	Sr	SonarQube
199	Pd	Hv	HiveMind
200	En	Wp	Web Services API
201	Os	Cm	Cloudbees AST
202	Fr	Bd	BlackDuck
203	Pd	Sr	SonarQube
204	En	Hv	HiveMind
205	Os	Wp	Web Services API
206	Fr	Cm	Cloudbees AST
207	Pd	Bd	BlackDuck
208	En	Sr	SonarQube
209	Os	Hv	HiveMind
210	Fr	Wp	Web Services API
211	Pd	Cm	Cloudbees AST
212	En	Bd	BlackDuck
213	Os	Sr	SonarQube
214	Fr	Hv	HiveMind
215	Pd	Wp	Web Services API
216	En	Cm	Cloudbees AST
217	Os	Bd	BlackDuck
218	Fr	Sr	SonarQube
219	Pd	Hv	HiveMind
220	En	Wp	Web Services API
221	Os	Cm	Cloudbees AST
222	Fr	Bd	BlackDuck
223	Pd	Sr	SonarQube
224	En	Hv	HiveMind
225	Os	Wp	Web Services API
226	Fr	Cm	Cloudbees AST
227	Pd	Bd	BlackDuck
228	En	Sr	SonarQube
229	Os	Hv	HiveMind
230	Fr	Wp	Web Services API
231	Pd	Cm	Cloudbees AST
232	En	Bd	BlackDuck
233	Os	Sr	SonarQube
234	Fr	Hv	HiveMind
235	Pd	Wp	Web Services API
236	En	Cm	Cloudbees AST
237	Os	Bd	BlackDuck
238	Fr	Sr	SonarQube
239	Pd	Hv	HiveMind
240	En	Wp	Web Services API
241	Os	Cm	Cloudbees AST
242	Fr	Bd	BlackDuck
243	Pd	Sr	SonarQube
244	En	Hv	HiveMind
245	Os	Wp	Web Services API
246	Fr	Cm	Cloudbees AST
247	Pd	Bd	BlackDuck
248	En	Sr	SonarQube
249	Os	Hv	HiveMind
250	Fr	Wp	Web Services API
251	Pd	Cm	Cloudbees AST
252	En	Bd	BlackDuck
253	Os	Sr	SonarQube
254	Fr	Hv	HiveMind
255	Pd	Wp	Web Services API
256	En	Cm	Cloudbees AST
257	Os	Bd	BlackDuck
258	Fr	Sr	SonarQube
259	Pd	Hv	HiveMind
260	En	Wp	Web Services API
261	Os	Cm	Cloudbees AST
262	Fr	Bd	BlackDuck
263	Pd	Sr	SonarQube
264	En	Hv	HiveMind
265	Os	Wp	Web Services API
266	Fr	Cm	Cloudbees AST
267	Pd	Bd	BlackDuck
268	En	Sr	SonarQube
269	Os	Hv	HiveMind
270	Fr	Wp	Web Services API
271	Pd	Cm	Cloudbees AST
272	En	Bd	BlackDuck
273	Os	Sr	SonarQube
274	Fr	Hv	HiveMind
275	Pd	Wp	Web Services API
276	En	Cm	Cloudbees AST
277	Os	Bd	BlackDuck
278	Fr	Sr	SonarQube
279	Pd	Hv	HiveMind
280	En	Wp	Web Services API
281	Os	Cm	Cloudbees AST
282	Fr	Bd	BlackDuck
283	Pd	Sr	SonarQube
284	En	Hv	HiveMind
285	Os	Wp	Web Services API
286	Fr	Cm	Cloudbees AST
287	Pd	Bd	BlackDuck
288	En	Sr	SonarQube
289	Os	Hv	HiveMind
290	Fr	Wp	Web Services API
291	Pd	Cm	Cloudbees AST
292	En	Bd	BlackDuck
293	Os	Sr	SonarQube
294	Fr	Hv	HiveMind
295	Pd	Wp	Web Services API
296	En	Cm	Cloudbees AST
297	Os	Bd	BlackDuck
298	Fr	Sr	SonarQube
299	Pd	Hv	HiveMind
300	En	Wp	Web Services API
301	Os	Cm	Cloudbees AST
302	Fr	Bd	BlackDuck
303	Pd	Sr	SonarQube
304	En	Hv	HiveMind
305	Os	Wp	Web Services API
306	Fr	Cm	Cloudbees AST
307	Pd	Bd	BlackDuck
308	En	Sr	SonarQube
309	Os	Hv	HiveMind
310	Fr	Wp	Web Services API
311	Pd	Cm	Cloudbees AST
312	En	Bd	BlackDuck
313	Os	Sr	SonarQube
314	Fr	Hv	HiveMind
315	Pd	Wp	Web Services API
316	En	Cm	Cloudbees AST
317	Os	Bd	BlackDuck
318	Fr	Sr	SonarQube
319	Pd	Hv	HiveMind
320	En	Wp	Web Services API
321	Os	Cm	Cloudbees AST
322	Fr	Bd	BlackDuck
323	Pd	Sr	SonarQube
324	En	Hv	HiveMind
325	Os	Wp	Web Services API
326	Fr	Cm	Cloudbees AST
327	Pd	Bd	BlackDuck
328	En	Sr	SonarQube
329	Os	Hv	HiveMind
330	Fr	Wp	Web Services API
331	Pd	Cm	Cloudbees AST
332	En	Bd	BlackDuck
333	Os	Sr	SonarQube
334	Fr	Hv	HiveMind
335	Pd	Wp	Web Services API
336	En	Cm	Cloudbees AST
337	Os	Bd	BlackDuck
338	Fr	Sr	SonarQube
339	Pd	Hv	HiveMind
340	En	Wp	Web Services API
341	Os	Cm	Cloudbees AST
342	Fr	Bd	BlackDuck
343	Pd	Sr	SonarQube
344	En	Hv	HiveMind
345	Os	Wp	Web Services API
346	Fr	Cm	Cloudbees AST
347	Pd	Bd	BlackDuck
348	En	Sr	SonarQube
349	Os	Hv	HiveMind
350	Fr	Wp	Web Services API
351	Pd	Cm	Cloudbees AST
352	En	Bd	BlackDuck
353	Os	Sr	SonarQube
354	Fr	Hv	HiveMind
355	Pd	Wp	Web Services API
356	En	Cm	Cloudbees AST
357	Os	Bd	BlackDuck
358	Fr	Sr	SonarQube
359	Pd	Hv	HiveMind
360	En	Wp	Web Services API
361	Os	Cm	Cloudbees AST
362	Fr	Bd	BlackDuck
363	Pd	Sr	SonarQube
364	En	Hv	HiveMind
365	Os	Wp	Web Services API
366	Fr	Cm	Cloudbees AST
367	Pd	Bd	BlackDuck
368	En	Sr	SonarQube
369	Os	Hv	HiveMind
370	Fr	Wp	

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

Key concepts:

- Extensive use of automation is a hallmark of DevOps
- Less turbulent releases that are faster and more predictable are only possible with automation
- The same can be said for reducing time to recover when errors do occur
- Technical staff should devote quality time to designing and deploying automation capabilities
- Automation also frees people to communicate and collaborate and to work on solving business challenges, improving processes (particularly overly bureaucratic processes), removing constraints and identifying, analyzing and mitigating risks
- Let systems manage systems and let people manage the business

The diagram illustrates the DevOps Toolchain as a continuous loop. It starts with **SOURCE CONTROL / VERSIONING** (represented by a person at a computer icon), which feeds into **CONTINUOUS INTEGRATION** (represented by a circular arrow icon). This integration step then leads to **CONFIGURATION MANAGEMENT** (represented by two interlocking gears icon). Finally, it returns to the **DEVELOPMENT ENVIRONMENT** (represented by a wrench and screwdriver icon). A large double-headed arrow labeled **INTEGRATION** connects the two main stages of the toolchain.

Up next

- How Netflix uses Jenkins** (Coding Techniques, 58K views)
- Top 10 DevOps Tools** (edureka!, 20K views)
- What is DevOps?** (edureka!, 891K views)
- Agile & DevOps** (John Okoro, 59K views)

The DevOps Toolchain with John Okoro (7:43)

Published on May 28, 2016 | Module 6: Automation & Architecting DevOps Toolchains | SUBSCRIBED 715 | 5:38

<https://youtu.be/bwE8aFPAzj8>

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

Module 6: Automation & Architecting DevOps Toolchains

149

Key concepts:

- These concepts and definitions are examinable

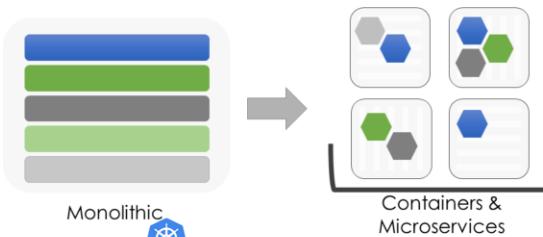
Resources:

- <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>

Cloud, Containers and Microservices



State of DevOps 2019
Elite performers were **24 times** more likely to have met all essential cloud characteristics than low performers.



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

Module 6: Automation & Architecting DevOps Toolchains

150

Key concepts:

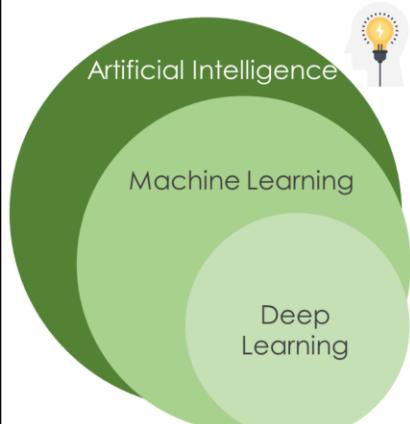
- Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP) are the most popular cloud platform providers
- Some people use the term ‘private cloud’ where IT services are provisioned over **private IT infrastructure** for the dedicated use of a single organization
- Docker is the most commonly used container solution
- Take some time to think about why cloud, containers and microservices are increasingly the standard architecture for applications (think little and often)
- Containers can hold applications rather than microservices and microservices don’t have to live in containers – but there are a lot of benefits to having microservices in containers

Resources:

- <https://techbeacon.com/enterprise-it/containers-microservices-modernizing-legacy-applications>
- <https://electric-cloud.com/resources/webinars/get-loose-microservices-and-loosely-coupled-architectures-featuring-jez-humble/>

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.

Module 6: Automation & Architecting DevOps Toolchains

151

Key Concepts:

- Artificial intelligence is a huge growth area and applied to data helps machines learn
- Applications are wide and varied across all industries
- How is your organization using AI and machine learning to improve value outcomes today?
- What plans or ideas do you have on how you can use your data better?

Resources:

- <https://techbeacon.com/enterprise-it/10-ways-machine-learning-can-optimize-devops>
- <https://devops.com/machine-learning-ai-driving-devops-evolution/>

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!

Key concepts:

- Limiting tool choices (standardization) can be a constraint and so a common DevOps practices is to integrate a collection of task-specific tools
- Start with an understanding of the artifact and information flow (e.g. via value stream mapping) and then begin looking at tools
- With teams collaborating and coordinating around the same tools, Dev and Ops teams can better understand each other's requirements and working methods
- Self-service can improve the timeliness of, for example, environment creation
- Test-driven development is a common practice in DevOps organizations and requires architecting software in a way that enables test automation
- Many organizations also require Dev to build the code that enables monitoring into each application as a standard practice
- Technical staff should devote quality time to devising and deploying automation capabilities
- Infrastructure as code is the practice of writing code (scripts) to provision and configure infrastructure

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

Key concepts:

- A variety of tools support communication and collaboration including full platforms that allow for multiple modes of communication from a single tool.
- Each organization must determine the tools it will use and the rules of engagement associated with those tools
- Real-time collaboration using common tools helps to improve flow
- ChatOps – a concept introduced by Git – involves using group chat rooms aided by a chat bot, Hubot that enables the automation (see article below for details)
- A shared backlog allows both teams to select improvement projects from an organizational (vs. local) perspective
- A shared backlog can be used to prioritize the work that delivers the greatest organizational value and helps to pay down technical debt

Resources:

- <https://devops.com/chatops-communicating-speed-devops/>
- <https://hubot.github.com/>

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.

Module 6: Automation & Architecting DevOps

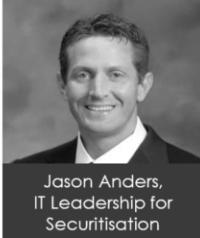
154

Key concepts:

- A good place to start is with repetitive tasks
- Every time a task comes up a second time, make a commitment organization wide to automate it – no matter how simple - and track changes in version control
- With each automated task you will free up time that can be invested in automating more complex tasks

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."



Jason Anders,
IT Leadership for
Securitisation

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

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

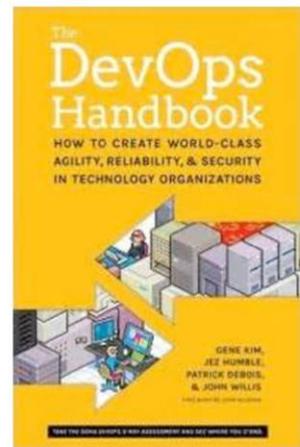
- <http://searchitoperations.techtarget.com/feature/Fannie-Mae-securitization-app-leads-DevOps-implementation>
- <https://dzone.com/articles/applying-devops-to-deliver-quality-at-speed>

DevOps Toolchains

Module 6: Automation & Architecting 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



Module 6: Automation & Architecting DevOps Toolchains

157

Key concepts:

- These shared services facilitate standardization, which enables engineers to quickly become productive, even if they switch between teams
- For instance, if every product team chooses a different toolchain, engineers may have to learn an entirely new set of technologies to do their work, putting the team goals ahead of the global goals

DISCUSSION

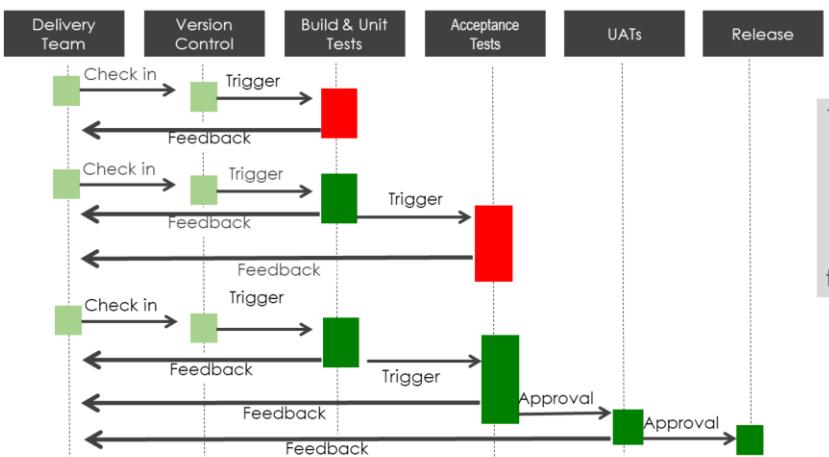
Applying the DevOps Handbook's Definition

Module 6: Automation & Architecting DevOps Toolchains

Think about the previous slide and discuss where you follow the definition (the learning facilitator may find it useful to write the second part as a list on a flipchart):

- Does IT Operations provide DevOps toolchains as a shared service in your organization?
 - What benefits and drawbacks would there be to this approach?
- Does your DevOps toolchain have:
 - A shared version control repository?
 - Pre-blessed security libraries? (Clue: If you have to ask what these are, you probably don't)
 - A deployment pipeline
 - Automated code quality
 - Automatic security scanning tools
 - Known, good environments
 - With production monitoring

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: ContinuousDelivery:
Reliable Software Releases
through Build, Test, and
Deployment Automation

Module 6: Automation & Architecting DevOps Toolchains

159

Key concepts:

- This graphic from Jez Humble and Dave Farley's book, 'Continuous Delivery' visualizes not only the stages of the deployment pipeline toolchain but also the underlying process
- Take some time to work through the steps

DevOps Toolchains

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

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

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

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

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

Module 6: Automation & Architecting DevOps Toolchains

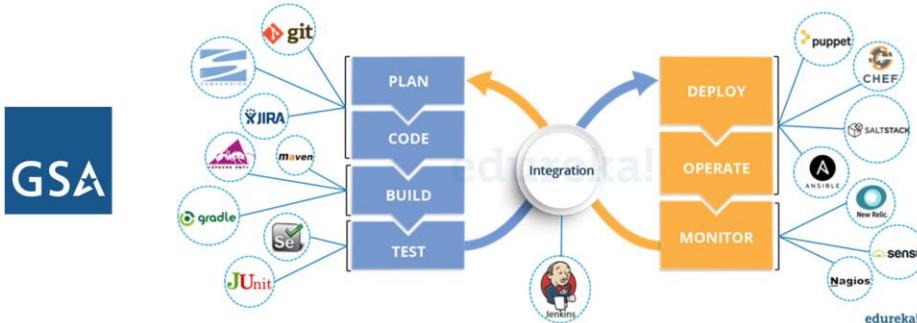
160

Key concepts:

- Definition source: Gartner - Avoid Failure by Developing a Toolchain That Enables DevOps (Williams, Murphy)
- At an abstract level, a deployment pipeline is an automated manifestation of your process for getting software from version control into the hands of your users (Jez Humble)
- **The deployment pipeline ensures that all code checked into version control is automatically built and tested in a production-like environment**
- This is also a good point to discuss the interfaces of the toolchain/deployment pipeline into administrative and/or support tools and processes (Incident, Problem, Event, Monitoring, etc.)
- The entire application stack and environment can be bundled into containers, which can enable unprecedented simplicity and speed across the entire deployment pipeline.
- **Technology ecosystems** are product platforms defined by core components made by the platform owner and complemented by applications made by autonomous companies in the periphery.

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.

Module 6: Automation & Architecting DevOps Toolchains

161

Key concepts:

- In the context of business, an ecosystem is a set of businesses functioning as a unit (e.g., a DevOps ecosystem) and interacting with a shared market for software and services
- Software ecosystems can also consist of a company providing a software platform and a community of external developers (and perhaps even external companies) providing functionality that extends the basic platform
- Selecting tools that are part of an existing ecosystem, web-based tools or tools that offer APIs allows the automation of one tool to kick off downstream work for the next tool
- An interface is a common boundary shared by two applications or programs that allow both to communicate with one another
- An API is a way for programmers to communicate with a certain application

Resources:

- https://tech.gsa.gov/guides/building_devsecops_culture/
- <https://www.sonatype.com/devsecops-reference-architectures>

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

Key concepts:

This slide describes the stages and common elements of a DevOps tool chain, usually built from multiple tools that are connected through APIs.

EXERCISE

Architect Your DevOps Toolchain

Module 6: Automation & Architecting DevOps Toolchains

Using a flipchart, describe your current DevOps toolchain(s) to the group. You may want to use different colored pens for:

- Tool category
- Tool name/brand
- Old/current and future or currently in PoC

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

Module 6: Automation & Architecting DevOps Toolchains

164

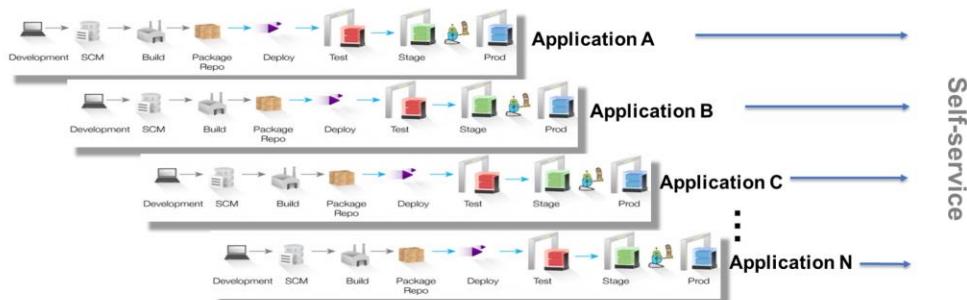
Key concepts:

This is a recommended sequence for introducing increasing levels of automation in the creation of a toolchain. Of course, some will happen simultaneously depending on the organization and in large organizations there are likely to be multiple toolchains being built by multiple teams at different speeds with different tools – at this point, the organization may wish to take the advice from the DevOps handbook and ask the IT operations team or form an infrastructure team to design and deliver a (variable) DevOps toolchain as a central service.

Resource:

- <http://www.informit.com/articles/article.aspx?p=1621865&seqNum=8>

Multiple Business Applications Require Multiple Toolchains



Source: Sanjeev Sharma, IBM

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

© 2015 IBM Corporation

Module 6: Automation & Architecting DevOps Toolchains

Key concepts:

While it may seem as if organizations have a single deployment pipeline and toolchain, the more likely scenario is that organizations have multiple deployment pipelines and toolchains for the various software products and services. The key is to ensure that the pipelines and toolchains are not operating in complete isolation, particularly since there may be a need to interface between them. The risk is of encouraging toolchain silos or conflicts in the testing and production environment.

Best practices for DevOps toolchains include:

- Provision by IT operations or an infrastructure squad
- Architected up front with standardized tool categories
- Use sensible defaults where possible (use common sense to decide which tools teams should choose when there are multiple on offer, based on what is already in use and liked by the organization as a whole)
- Make available to be consumed via self-service
- Note that these toolchains are not necessarily the same – they have the same categories but they may have different tools

Module 6: Quiz

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

Module 6: Quiz Answers

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

Module 7

MEASUREMENT, METRICS & REPORTING

Module 7: Measurement, Metrics & Reporting

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

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

The Importance of Measurement



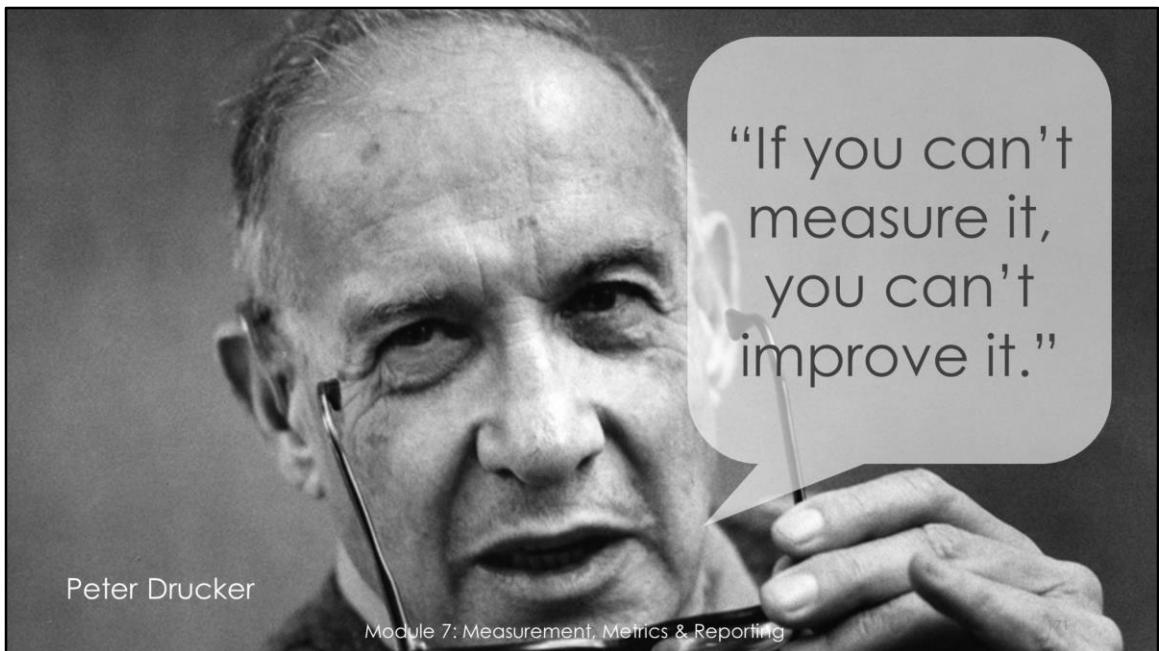
The First Way	The Second Way	The Third Way
Flow	Feedback	Continuous Experimentation & Learning
<ul style="list-style-type: none"> • Change lead time • Change cycle time • Time to value • Value realisation 	<ul style="list-style-type: none"> • Build/test results • Change fail rate • Monitoring • % rework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

Key Concepts:

- You may find it useful to think of measurement through The Three Ways framework
- Think about which kinds of metrics tell us more about flow and experimentation/learning than feedback for example

Resources:

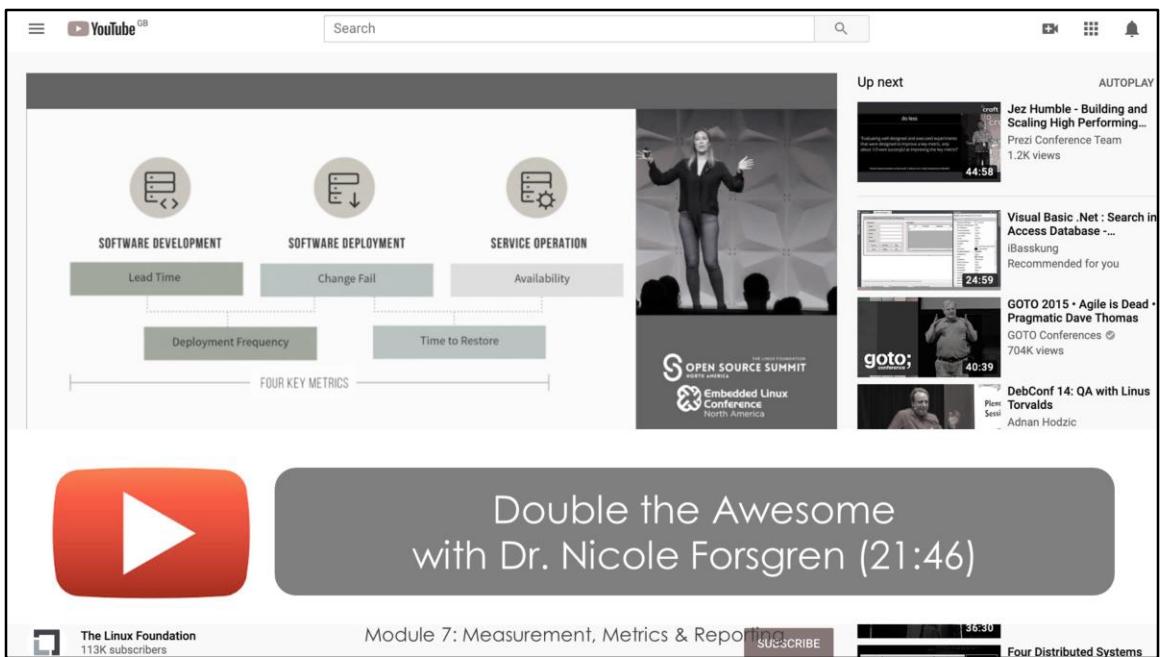
- <https://www.linkedin.com/pulse/devops-measure-value-speed-stephen-williams/>



Peter Drucker

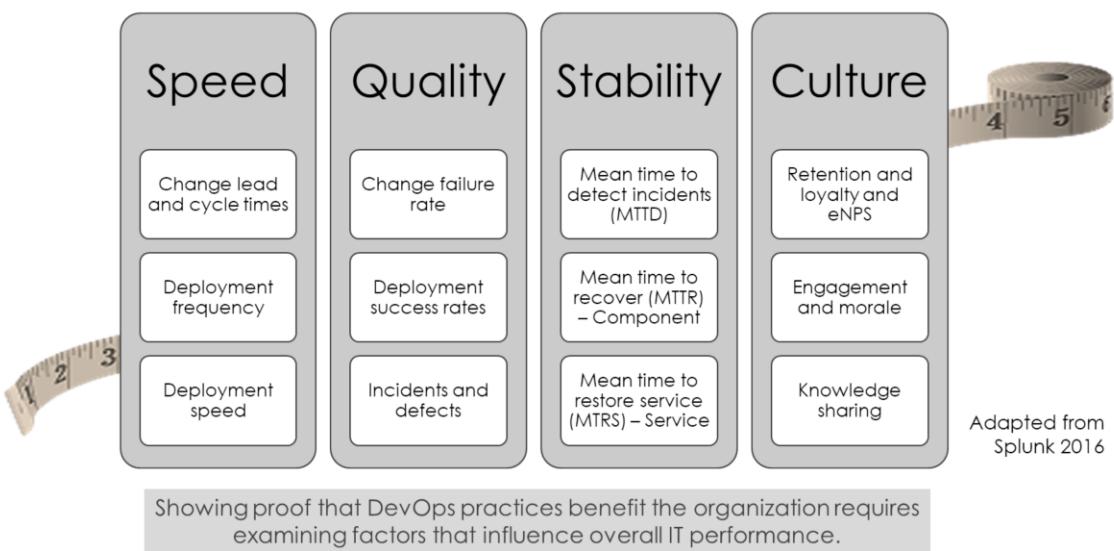
Module 7: Measurement, Metrics & Reporting

171



<https://youtu.be/c2VwRKAyQ-M>

Measuring Success



Adapted from Splunk 2016

Module 7: Measurement, Metrics & Reporting

173

Key concepts:

- This slide shows just some of the key metrics organizations can use to show the effectiveness of their DevOps practices
- Showing proof that DevOps practices benefit the organization requires examining factors that influence overall IT performance
- In other words, you can't measure Dev and Ops separately
- Showing proof that DevOps practices benefit the organization also requires baselining key metrics before and then after making improvements
- Change lead time is one of the most important metrics as it represents what the customer sees
- Cycle time is a more mechanical measure of process capability
- Lead time depends on cycle time, but also depends on your willingness to keep a backlog, the customer's patience, and the customer's readiness for delivery
- Change failure rate is an important measure of reliability and stability
- Note that people sometimes think that MTTR and MTRS can be used interchangeably
- MTTR is measured from when a component fails until it is repaired
- MTTR does not include the time required to recover or restore service (e.g., data may need to be recovered before service is fully restored and delivering its normal functionality)
- While it may be hard to assign a dollar value, there are ways to measure culture
- Representative metrics include: stress levels, collaboration levels, the ability to attract and retain talent, turnover, employee morale, responsiveness to change, cross skilling, knowledge sharing, resource utilization
- Employee Net Promoter Score (eNPS) is a way for organizations to measure employee loyalty. The Net Promoter Score, originally a customer service tool, was later used internally on employees instead of customers

Resources:

- http://devopsentreprise.io/media/DOES_forum_metrics_102015.pdf
- <https://codeascraft.com/2010/12/08/track-every-release/>

DISCUSSION

Metrics Used Today

Think about the previous slide and discuss where:

- Which of these metrics you currently measure
- How and when you measure them
- To whom they are reported and when
- What actions are taken as a result of the measurements

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

Module 7: Measurement, Metrics & Reporting

175

Key Concepts:

- Many metrics are measured differently from organization to organization and even team to team within an organization
- It's important to form and agree a common language and set of definitions so that metrics can be shared cross-team
- A key metric for the technology teams to provide to the business is change/lead or cycle time – how long it takes them to get the thing that they've asked for
- People need to agree whether there is a difference between lead and cycle time and when to start measuring from
- This metric can highlight the differences between a waterfall or waterfall – like approach versus an agile approach
- This differential definition from Accelerate refers to the agile environment

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

Module 7: Measurement, Metrics & Reporting

176

Key Concepts:

- Most organizations will have none/virtually no metrics or so many metrics that they have become meaningless and will need to place new focus on identifying the ones that are key – with the acknowledgement that they will change
- When you are planning your metrics, consider this advice carefully:
 - Focus on outcomes and value
 - Capability is better than maturity because we are never ‘done’
 - Lines of code are dangerous to measure because often, the simpler solution is the more elegant and the less likely to result in technical debt; as Da Vinci said; “Simplicity is the ultimate sophistication”
 - Measure velocity in an agile team to help with estimation and cadence – but don’t compare velocity across teams – since they will be doing things in (slightly) different ways this is dangerous and likely to drive the wrong sorts of behaviors
 - Measuring purely on utilization can miss on embedding experimentation and learning in your culture, may not tackle technical debt and risks burnout culture
 - Team and global metrics help drive visibility and transparency and consequently collaboration, shared accountability, goals and vision

- Do you measure unplanned work?

Resources:

- <https://itrevolution.com/book/accelerate/>
- <http://cuddletech.com/slides/DevOps-Demystified.pdf>

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

Resource:

- <https://blog.xebialabs.com/2016/11/15/reap-rewards-devops-one-banks-story/>

Gartner DevOps Metrics Pyramid



Module 7: Measurement, Metrics & Reporting

178

Key concepts:

- This pyramid for Gartner isn't a maturity pyramid – you don't aim for the top
- It's a hierarchy of 'importance' of the metrics placed on it and elevates the business metrics
- Take some time to review the pyramid considering:
 - Which of the metrics you currently measure
 - Which of those you think would be useful for your organization or not

Resource:

<https://www.gartner.com/doc/3760663/datadriven-devops-use-metrics-guide>

EXERCISE

The Most Meaningful Metrics

Take 10 minutes to identify:

- 1) The top three metrics you are currently asked to measure and report on – be exact about how you define ‘top’
- 2) The one metric you think is most valuable to your business, why, how you measure it – this does not have to be a metric you currently measure
- 3) What outcomes you receive or expect to receive by using this metric

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 |

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 |

Module 8

SHARING, SHADOWING & EVOLVING

Module 8: Sharing, Shadowing & Evolving

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

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

DevOps Encourages a Sharing Culture

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

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

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

Module 8: Sharing, Shadowing & Evolving

184

Key Concepts:

Actively sharing tools, knowledge, discoveries and lessons learned helps Dev and Ops to:

- Identify new collaboration opportunities
- Avoid redundant work and overcome silo cultures
- Create common vocabularies and mindsets
- Create active exchanges of ideas and innovation
- Respect each other's skills, expertise and commitment

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

youtube.com

Up next

AUTOPLAY

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 Sil English Rackspace 891K views 7:07

CHARLIE MUNGER Charlie Munger Interview 2018

Module 8: Sharing, Shadowing & Evolving

Puppet Published on Apr 25, 2016

RECOMMENDED FOR YOU

9:49

<https://youtu.be/8aJHtp--3U>

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

Key Concepts:

- An "open space" is where attendees have an opportunity to propose topics and those interested in those topics can join in discussions that are relevant to them
- It's the TwoFeet approach where an attendee's feet can take them where they want to be and an example of self-organization in a conference setting
- If you were having a DevOps Days session in your organization and you got to the Open Spaces part of the day, what topic would you propose?

Resource:

- <https://www.devopsdays.org/open-space-format/>

DISCUSSION

What's your Open Space topic?

Module 8: Sharing, Shadowing & Evolving

Take a few moments to think about which of the topics covered over the last couple of days you would most like to explore further and would propose as an Open Space discussion if you were having a DevOps Days at your organization. Share your idea with the group and see if there are any trends or patterns.

DevOps in the Enterprise

Module 8: Sharing, Shadowing & Evolving

CASE STORY: Disney

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

“The digital expansion of business means more work and firefighting.”



Jason Cox,
Director of Systems
Engineering

Benefits

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

- <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>

Roles

Module 8: Sharing, Shadowing & Evolving

Addressing the DevOps Skills Gap

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

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

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

Key Concepts:

- The talent to fill needed roles may be:
 - Within enterprise IT
 - Within shadow IT groups
 - Outside the organization
- Some organizations hire or contract talent from outside the IT organization to jump start DevOps initiatives

Resources:

- <https://techbeacon.com/devops/6-devops-recruiting-tips-how-land-right-people>
- <https://www.cloudcomputing-news.net/news/2016/jun/16/devops-engineer-hardest-job-to-fill/>
- <https://www.upguard.com/blog/devops-new-job-skills-roles-titles>

Differences of Perspectives of “Must-Have”

Skill Category	C-Suite Selected “Must-Have”	Management Selected “Must-Have”	Individual Contributor “Must-Have”
Process Skills and Knowledge	54%	58%	50%
Automation Skills	71%	58%	56%
Functional Knowledge	31%	35%	44%
Specific Automation Tools	29%	30%	20%
Specific Certifications	14%	16%	8%
Business Skills	43%	30%	22%
Soft Skills	69%	55%	44%

2019

Upskilling:
Enterprise DevOps Skills Report
Presented by DevOps Institute

Module 8: Sharing, Shadowing & Evolving

192

Key Concepts:

- DevOps practitioners do not all do the same things; however, the digital era introduces a set of continually evolving challenges across not just the technology topic, but also relative to the demographics of employees and customers.
- As millennials overtake baby boomers by 2019 in the United States and companies continually working on improving customer, employee and partner experience the adoption of DevOps will only rise.
- While DevOps needs particular champions, excellence can be achieved only if the entire organization is on board, and that requires constant evangelism and training
- But what are the key skills to look for when recruiting?
- What are the key skills to grow when you want to join the DevOps movement either inside or outside your company
- To help you DOI have further analyzed the perspectives of the survey takers and their priorities across the following skill categories: automation, process, soft skills, functional, specific automation tools, business skills and certification
- This slide shows the priorities across the top skill categories relative to the key roles surveyed

Resource:

- <https://info.devopsinstitute.com/upskilling-report-2019>

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.

Key Concepts:

- The demand for DevOps skills is rapidly rising
- These lists aren't intended to be all inclusive; rather, they are a collection of the most commonly-referenced DevOps skills and characteristics
- Note that 'T-shaped' refers to someone who has depth of related skills and expertise in a single field and who also has broader skills and knowledge and learns by linking up different perspectives from different specialties
- Note also that 'writing code' doesn't relate only to software development
- It also includes developing automated tests, deployment scripts, etc.
- While experience with specific tools may be desirable, many organizations provide training on their specific tools

Resources:

- <https://techbeacon.com/devops/7-keys-finding-phenomenal-devops-talent>
- <https://devops.com/devops-7-skills-ops-pros-need-succeed/>
- [https://www.theregister.co.uk/2015/11/02/t shaped developers are the new n
ormal/](https://www.theregister.co.uk/2015/11/02/t_shaped_developers_are_the_new_normal/)

DevOps Roles

What other roles do you think should be involved?

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

Module 8: Sharing, Shadowing & Evolving

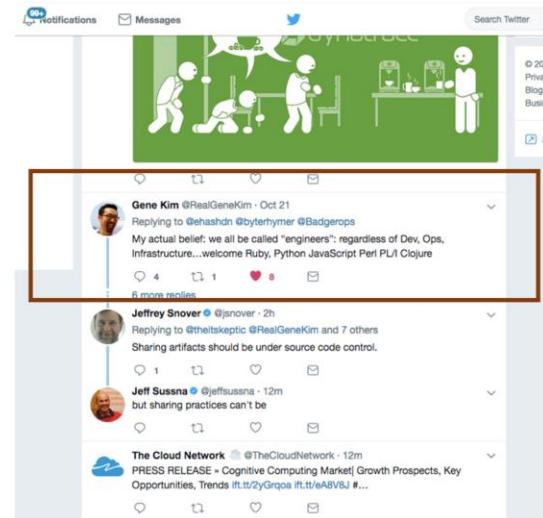
194

Resource:

- <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



Key concepts:

- While this is not an ‘industry recognized’ role, there are job posting all over the internet, some of which are extremely comprehensive
- DevOps Engineers are often viewed as the people who sit between or within Development and Operations and facilitate the use of technology to help their organizations deliver better software, faster and more reliably.
- This role is not, however, without its controversy since there is not a standard job description or set of roles and responsibilities
- Many leading proponents in the DevOps space believe we should not have job titles, just roles, and that we should ‘all be engineers’ (see Twitter quote from Gene Kim)

Resources:

- <https://puppet.com/blog/what-a-devops-engineer>
- <http://java.dzone.com/articles/just-what-devops-engineer>

DevOps Leadership

Module 8: Sharing, Shadowing & Evolving

Transformational Leadership

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

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

Dimensions of transformational leadership



The characteristics of transformational leadership are highly correlated with IT performance and employee Net Promoter Score (eNPS).
From The State of DevOps Report 2017

Module 8: Sharing, Shadowing & Evolving

197

Key Concepts:

- Significant differences in leadership characteristics were observed between high-, medium- and low- performing IT teams
- High-performing teams reported having leaders with the strongest behaviors across all dimensions
- Teams with the least transformative leaders (the bottom third) were also far less likely to be high IT performers - in fact, they were half as likely to exhibit high IT performance
- Though we often hear stories of DevOps and technology- transformation success coming from the grassroots, it is far easier to achieve success when you have effective leaders lending their support
- Transformational leadership is highly correlated with employee Net Promoter Score (eNPS) - transformational leaders in places where employees are happy, loyal, and engaged
- Employee Net Promoter Score (eNPS) is a way for organizations to measure employee loyalty. The Net Promoter Score, originally a customer service tool, was later used internally on employees instead of customers

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

<https://thenewstack.io/magic-behind-disney-devops-experience/>

Organizational Considerations

Module 8: Sharing, Shadowing & Evolving

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.

Key Concepts:

- Deployment frequency increases the need for communication and coordination increases as well
- Because they typically require a great number of handoffs, organizations with function-oriented may struggle to maintain the required higher rate of flow
- Some organizations address this need for greater collaboration by designating an Ops liaison for application or service teams
- These liaisons attend Dev standups and bring their needs back to Ops
- Function-oriented structures optimize for expertise and division of labor
- Market-oriented structures optimize for responding quickly to customer needs
- Matrix organizations attempt to balance the needs of function- and market-oriented structures by sharing resources
- Matrix organizations can be complex and can create conflict if efforts aren't made to reduce ambiguity in terms of peoples' priorities and reporting structure
- Last bullet - optimally these services are self-service
- Examples of services include enabling project teams to be quickly on boarded, leveraging common source code repositories, automated build and testing pipelines, monitoring tools, security scanning and tools, and so forth

DevOps Teams (1)

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



DevOps teams:

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

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

Key Concepts:

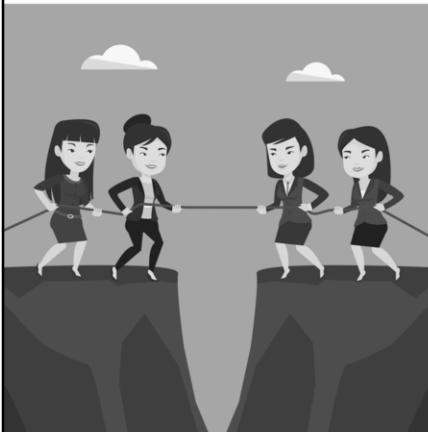
- Note that for the purposes of the exam there are pros and cons to creating a DevOps Team
- The number of respondents who indicated they are part of a DevOps department is up from 16% in 2014 and 19% in 2015
- For many organizations what is enabling these teams to work is that they are cross-functional teams organized around products or services, vs. projects
- In some organizations, the DevOps team's responsibility is to help ensure the smooth running of the source control repository, build systems and to help build the automated environments for testing and development to perform their daily tasks
- Dedicated DevOps teams are often made up of experienced operations people with a mix of skills including using version control, writing infrastructure as code, and continuous delivery (Puppet Labs)

Resources:

- <http://caseywest.com/the-mission-of-a-devops-team/>
- <http://www.michaelnygard.com/blog/2016/01/whats-lost-with-a-devops-team/>

- <http://searchitoperations.techtarget.com/feature/How-to-implement-a-solid-DevOps-team-structure>
- <https://continuousdelivery.com/2012/10/theres-no-such-thing-as-a-devops-team/>

DevOps Teams (2)



Downsides of dedicated DevOps Teams:

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

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

Key concepts:

- DevOps must be *everyone's* responsibility, not just those who are part of a DevOps team
- Introducing a DevOps team could lead to yet another silo
- Dev and Ops may now be even further removed from each other
- Collaboration, learning and sharing are stifled

Resource:

- <https://www.infoq.com/news/2017/10/devops-teams-good-or-bad>

Getting Started

Module 8: Sharing, Shadowing & Evolving

“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 Sargent



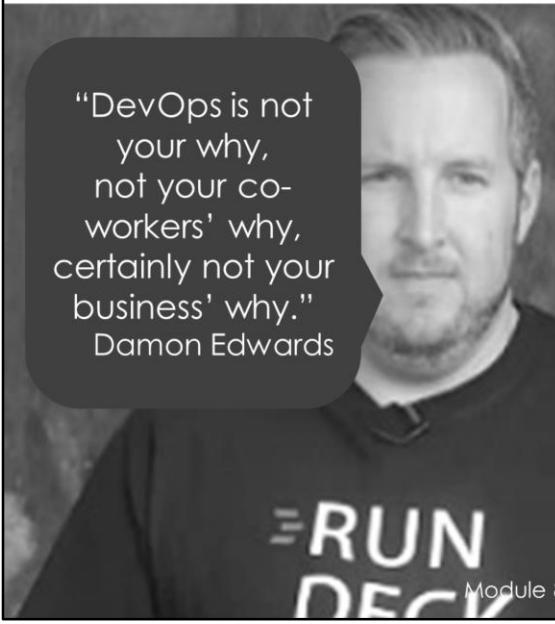
Module 8 Sharing, Shadowing & Evolving

204

Key concepts:

- Melissa Sargent is a Senior Director of Project and Solutions Marketing with CA Technologies

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

Key concepts:

This slide builds on the Kotter concepts and encourages organizations to start where they are today and build forward gradually as time, costs and business requirements dictate.

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

Key concepts:

- Briefly discuss considerations for selecting a pilot
- Ask learners if they have performed a pilot and what results have become apparent
- If not, ask them where they would consider a pilot in their organization, emphasizing that the best approach is a small one with friendly, understanding business customers

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

Module 8: Sharing, Shadowing & Evolving

207

Key concepts:

- Making available reusable artifacts and measurements is key
- Avoid requiring others to reinvent the wheel
- Build momentum and go faster with every improvement iteration by building on what's been done before
- Start simple at get more sophisticated as you go
- You can't get from level 1 maturity to level 5 without going through levels 2, 3 and 4

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

Key concepts:

- Culture changes only after people see that they new ways of doing things is better
- Keys to proving the new way is better include:
 - Ensuring that baselines are captured before you start so you can show measurable benefits (tangible)
 - Using surveys and feedback to capture people's perceptions (intangible)
- Know that some of your naysayers may move on and that's okay
- You want to incent and reward the people who are willing to embrace the new way of doing things

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

Module 8: Sharing, Shadowing & Evolving

210

Key concepts:

- These CSFs are common to any type of culture change
- Is management commitment enough? What about staff commitment?
- Also discuss how cultural changes and automation increases can be balanced
- Emphasize the need for general (Foundation) and subject matter expertise (cultural, principles, architecture and specific applications)
- Training also encourages the development of a common vocabulary and understanding as well as facilitates cultural discussions that might not otherwise happen

Challenges and Risks

Overcoming these challenges will require organizational change.

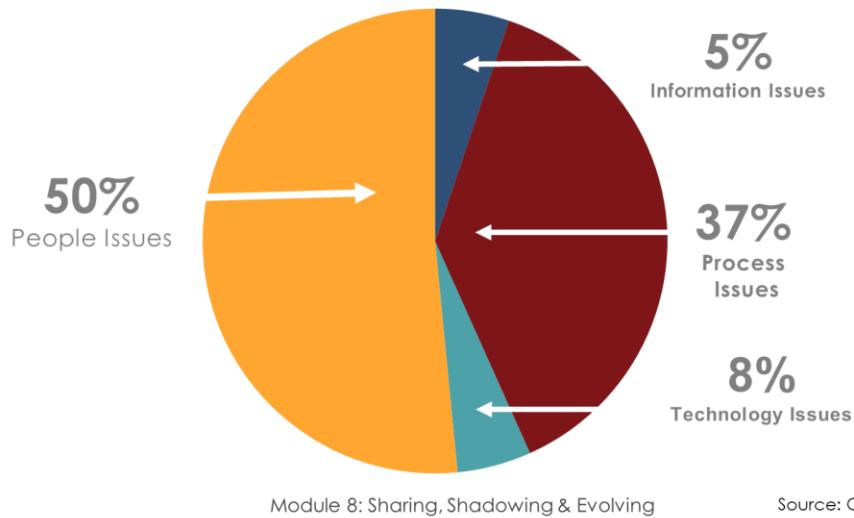


- Lack of commitment or clarity
- Transforming a “them” and “us” culture
- Blending teams that are geographically dispersed, unfamiliar with each other and may include suppliers
- Lack of education, training and skill
- Immature service management processes
- Inadequate technologies
- Poor communication

Key concepts:

- These challenges and risks are common to any type of culture change
- Take some time to consider which ones may apply to you and what you can do to address and mitigate them

Your Biggest Challenge for the Expansion of DevOps?



Key concepts:

- This infographic from Gartner demonstrates that many of the challenges are common between organizations
- It also shows the importance of investing in people, culture, learning and development in order to make DevOps a reality

EXERCISE

Your DevOps Experiment (Personal Action Plan)

Remember the Improvement Kata? Plan, Do, Check. Act

Write an experiment that you commit to undertaking:

Plan/Do: I believe that if I do this (quantified, specific, viewable) thing

Check: On this date I will be able to see THIS OUTCOME (quantified, viewable, specific)

Act

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 |

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 |

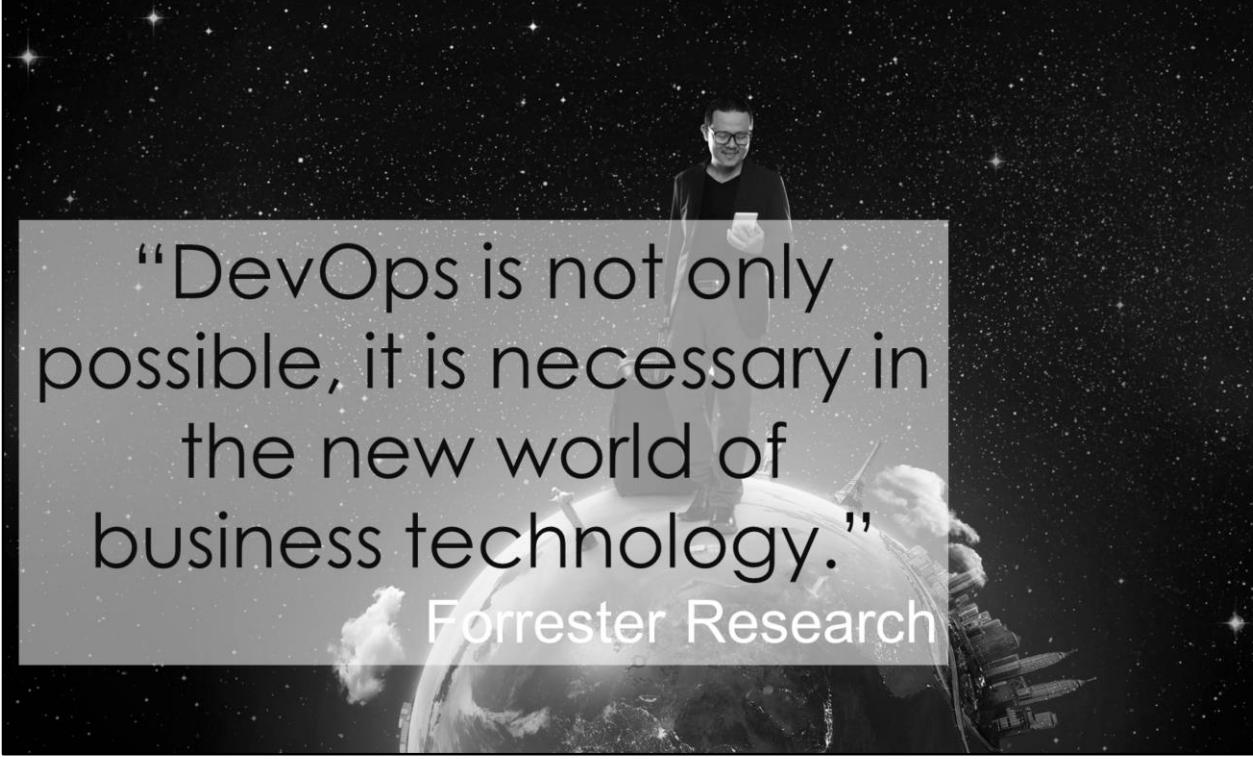
Summary

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

Key concepts:

This slide summarizes the "story" that sits behind the course.



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

Forrester Research

You Are Part of the DevOps Member Association

Join The DevOps Member Association

Advancing the Human Elements of DevOps



SKIL | FRAMEWORK
by the DevOps Institute



Skills
Career Growth
Certifications
Learning Paths
Job Board



Knowledge
Insight
Research Reports
eBooks
Articles



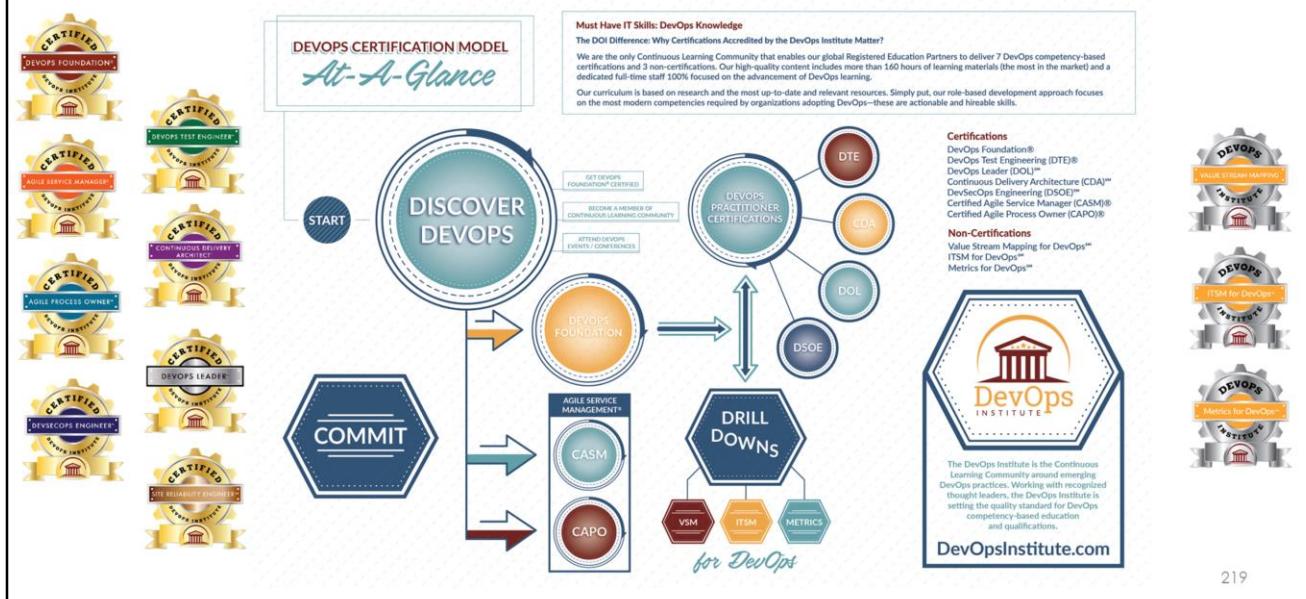
Ideas
Innovation
Forums
Events
Virtual Meetups



Learning
Know How
Videos
Micro Learning
Workshops

218

DOI Learning Tracks & Community



Key concepts:

The entry points the DOFD leads to and the DOI learning community