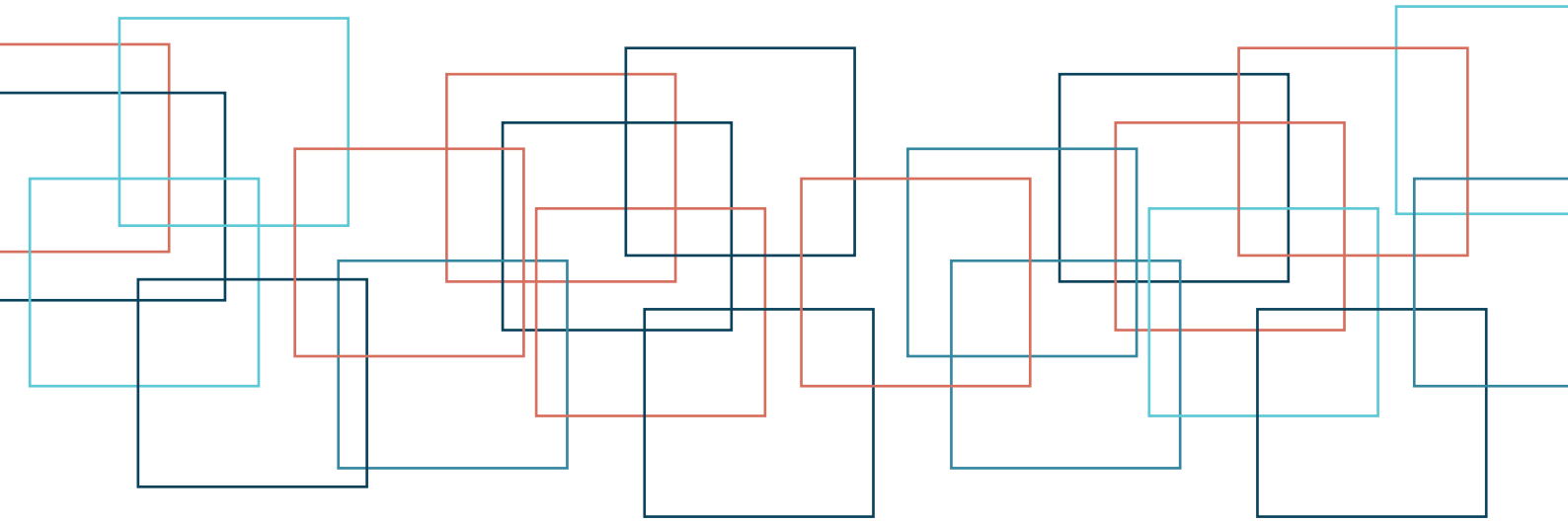


DevOps Fundamentals Select



Course Book

DevOps Fundamentals Select

Copyright and Disclaimer

DevOps Fundamentals Select | r1.3.0

Copyright

Copyright © 2017, ITpreneurs Nederland B.V. All rights reserved.

Copyright © Devops Agile Skills Association LLC 2017. All rights reserved.

Published under license by ITpreneurs Nederland B.V.

This is a commercial confidential publication. All rights reserved. This document may not, in a whole or in part, be copied, reproduced, translated, photocopied, or reduced to any medium without prior and express written consent from the publisher.

For permission requests, write to the publisher, addressed “Attention: Course Permissions,” at servicedesk@itpreneurs.com.

This course includes copyrightable work provided to ITpreneurs under license and is protected by copyright. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law or further disseminated without the express and written permission of the legal holder of that particular copyright. The Publisher reserves the right to revoke that permission at any time. Permission is not given for any commercial use or sale of this material.

Trade Marks

DevOps Fundamentals® is a registered trademark of DASA Limited.

Disclaimer

Information provided about the course, modules, topics and any services for courses including simulations or handouts, are an expression of intent only and are not to be taken as a firm offer or undertaking. The Publisher reserves the right to discontinue or vary or maintain such course, modules, topics, or services at any time without notice and to impose limitations on enrolment in any course.

The course materials provided may have hypertext links to a number of other web sites as a reference to users. This service does not mean that the publisher endorses those sites or material on them in any way. The publisher is not responsible for the use of a hypertext link for which a commercial charge applies. Individual users are responsible for any charges that their use may incur.

The information in this course is written using a blend of British and American English. Although every effort has been made regarding the usage of correct spelling, punctuation, vocabulary, and grammar with regard to the Standard English, ITpreneurs accepts no responsibility for any loss or inconvenience caused due to the regional differences in the usage of the English language.

Contents

ACKNOWLEDGEMENTS	V
MODULE 1: COURSE INTRODUCTION	1
Let's Get to Know Each Other	1
Overview	1
DASA Competence Framework	2
Course Objectives	3
DASA: Mastering DevOps in Practice (Knowledge and Skills)	4
Course Agenda	5
Type of Activities	5
Exam	6
Course Book	6
Technical Glossary	6
Question and Answer	7
Module Summary	7
Activity: Group Discussion	7
MODULE 2: DEVOPS INTRODUCTION	9
Module Objectives	9
Module Topics	10
Emergence of Devops	10
Activity: Group Discussion	11
Core Concepts of Devops	18
Activity: Group Discussion	21
Activity: Group Discussion	26
Module Summary	27
MODULE 3: CULTURE	29
Module Objectives	29
Module Topics	30
Essence of a Devops Culture	30
Key Elements of Devops	34
Implementation of a Devops Culture	50
Activity: Group Discussion	53
Module Summary	54
MODULE 4: ORGANIZATION	55
Module Objectives	55
Module Topics	56
Organizational Model	56
Autonomous Teams	61
Activity: Group Discussion	66
Architecting for Devops	66
Governance	73
Module Summary	75
MODULE 5: PROCESSES	77
Module Objectives	77
Module Topics	78
Process Basics	78
Devops in Relation to ITSM	79
Activity: Group Discussion	82
Agile and Scrum	82

Activity: Task Switching/Game	86
Optimizing Processes Using Lean	96
Business Value Optimization and Business Analysis Using Story Mapping	105
Module Summary	109
MODULE 6: AUTOMATION	111
Module Objectives	111
6A Automation Concepts Topics	112
6A Automation Concepts: Automation for Delivery of Software	112
6A Automation Concepts: Continuous Delivery Core Concepts	116
6A Automation Concepts: continuous Delivery Automation Concepts	121
6A Automation Concepts: Continuous Delivery Automation Focus Topics	126
6B Data Center Automation Topics	134
6B Data Center Automation: Cloud Technology and Principles	134
6B Data Center Automation: Cloud Services Concepts in a Devops Organization	136
6B Data Center Automation: Automated Provisioning Concepts	139
6B Data Center Automation: Platform Product Characteristics and Application Maturity	144
Module Summary	145
Activity: Group Discussion	145
MODULE 7: MEASURE AND IMPROVEMENT	147
Module Objectives	147
Module Topics	148
Importance of Measurement	148
Choosing the Right Metrics	150
Monitoring and Logging	157
Module Summary	161
EXAM PREPARATION GUIDE	163
MOCK EXAM	167
APPENDIX A: ANSWERS	187
APPENDIX B: SYLLABUS	191
APPENDIX C: GLOSSARY	211
APPENDIX D: RELEASE NOTES	221
APPENDIX E: PARTICIPANT FEEDBACK FORM	223

Acknowledgements

We would like to sincerely thank the experts who have contributed to the development of the DevOps Fundamentals Select.

Lead Author



Rik Farenhorst, Business Unit Manager at Xebia

Rik has a PhD in software engineering and more than 10 years of experience in the business-IT domain. He has worked as Enterprise/IT Architect, Consultant, Trainer and Coach, and has later grown into sr. management positions. He guides organizations and individuals in raising their bar by making IT simpler, better, and of higher quality. Rik works as Business Unit Manager at Xebia, a specialized international IT consultancy, solutions and training company specialized in high-performance IT, digital transformation, and thought leader in DevOps. Rik is a member of the editorial board of the DevOps Agile Skills Association (DASA).



Michiel Sens, Solution Architect at Xebia

Michiel Sens has worked in the IT Industry for over 18 years, is co-writer of the book 'Continuous Delivery for Managers' and works as a Principal Consultant/ Solution Architect at Xebia in The Netherlands. Michiel specializes in Continuous Delivery and full lifecycle software development programs and as such focuses on coaching Agile Teams, setting up central build environments, automating tests, automating deployment over multiple environments and implementing fully automated PaaS platforms. He advocates the use of Continuous Delivery at seminars and meetups and remains in touch with details by performing actual implementations as well.



Frederik Schukken, Managing Consultant at Quint Wellington Redwood

Frederik is a Managing Consultant at Quint Wellington Redwood where he and his team help businesses to adopt disruptive thinking and create high-performance IT. He is passionate about helping others to adopt Lean and DevOps methodologies and break out of rigid, traditional ways of working.



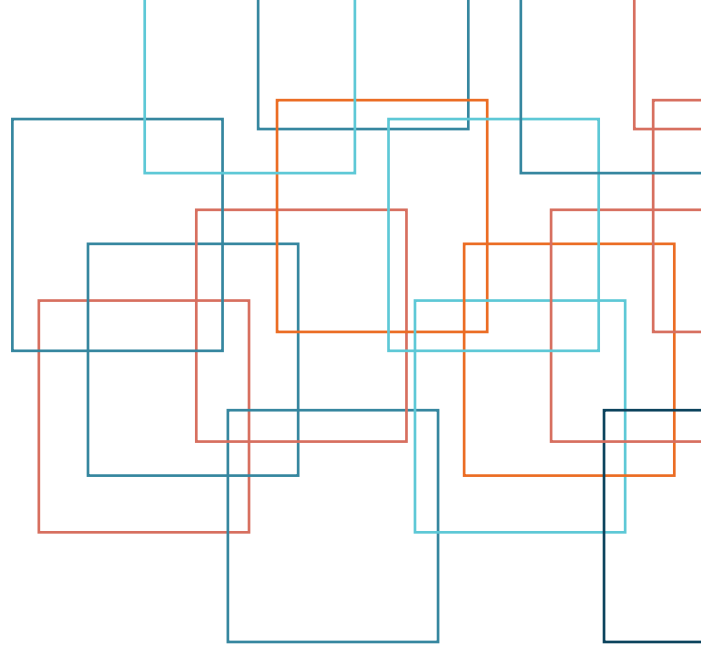
Thomas Kruitbosch, DevOps Consultant at Xebia

Thomas is specialized in Continuous Delivery and optimizing IT for fast adoption of new technologies. Thomas has extensive experience and in-depth knowledge of both development and operations. As Consultant, Trainer, and Conference Speaker, Thomas advocates the use of Agile and Automation Delivery principles. Also, he performs technical implementations of Continuous Delivery Automation and (cloud) platforms.



Niels Loader, Managing Consultant at Quint Wellington Redwood

As an advisor to 10 of IT organizations, Niels has extensive knowledge and experience in implementing IT Service Management, IT Performance Management, Lean IT, and DevOps within IT organizations. In 2010 and 2011, he was one of the initiators of the Lean IT Foundation certification and spent four years as the Chief Examiner for the APMG Lean IT certification. He is the Lead of the Content team of the Lean IT Association.



COURSE INTRODUCTION

Let's Get to Know Each Other

Introduce yourself in the following format:

- Name
- Company
- Role and background
- Familiarity with DevOps concepts and their practice
- Experience in application development, infrastructure development, and/or operations
- Expectations from this course

Overview

The topic is presented in the class through a video.

IT industry and software development are going through rapid changes. IT organizations are focusing on the customer-centric approach and delivering faster and better products. This focus is being brought about by DevOps. DevOps is being widely adopted by both, small businesses and large enterprises.

DevOps was born as an answer to resolve the prevalent conflicts between the Development and Operations. In traditional IT organizations, Development and Operations worked with conflicting interests and lack of communication and this often led to challenges such as slow releases, low quality, and high costs. DevOps aim to bring

the two worlds together and improve collaboration and communication between Development and Operations to improve efficiency and productivity.

But DevOps is much more than this. DevOps brings to the table increased agility and continuous delivery. The foundation of DevOps are Agile and Lean methods. The Agile methodology helps you to shorten the cycle times and enable frequent releases but DevOps is one step ahead of this. The Agile methodology focuses on developing products through continuous development cycle, but shows no focus on deployment. DevOps showed the way for achieving the common goal of continuous development as well as deployment.

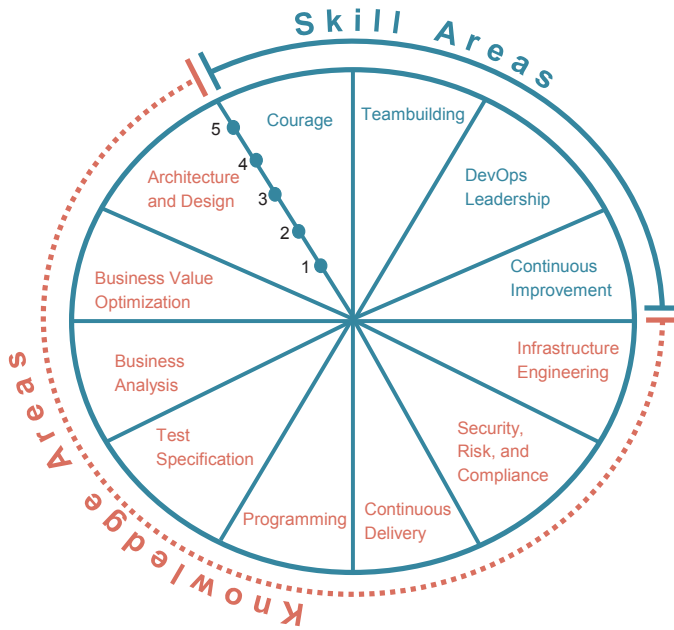
For accelerating continuous delivery, DevOps encourages automation of all development and deployment processes, including integration, testing and releasing. Automation and monitoring are among the principle characteristics of DevOps. Another important principle of DevOps is continuous improvement, which is based on the Lean principles that encourage feedback from customers to improve continuously.

Although, there is no established definition of DevOps, you can undoubtedly say that DevOps is a cultural and organizational movement that focuses on collaboration, team building, continuous delivery, automation and continuous improvement.

DevOps is a buzzword today and offer lucrative and exciting career opportunities. The 3-days classroom training on DevOps Fundamentals will not only introduce you to the basics of DevOps but will prepare you for a successful career. This training is designed to provide the core education necessary to build your DevOps vocabulary and understand DevOps principles and practices. You will be exposed to the key DevOps concepts and terminology through presentations , videos, group discussions, and activities. As the part of course material, you will receive a Course Book and an additional reference ebook. This course material provides you with all essential learning to earn your DevOps Fundamentals certification from DASA.

DASA Competence Framework

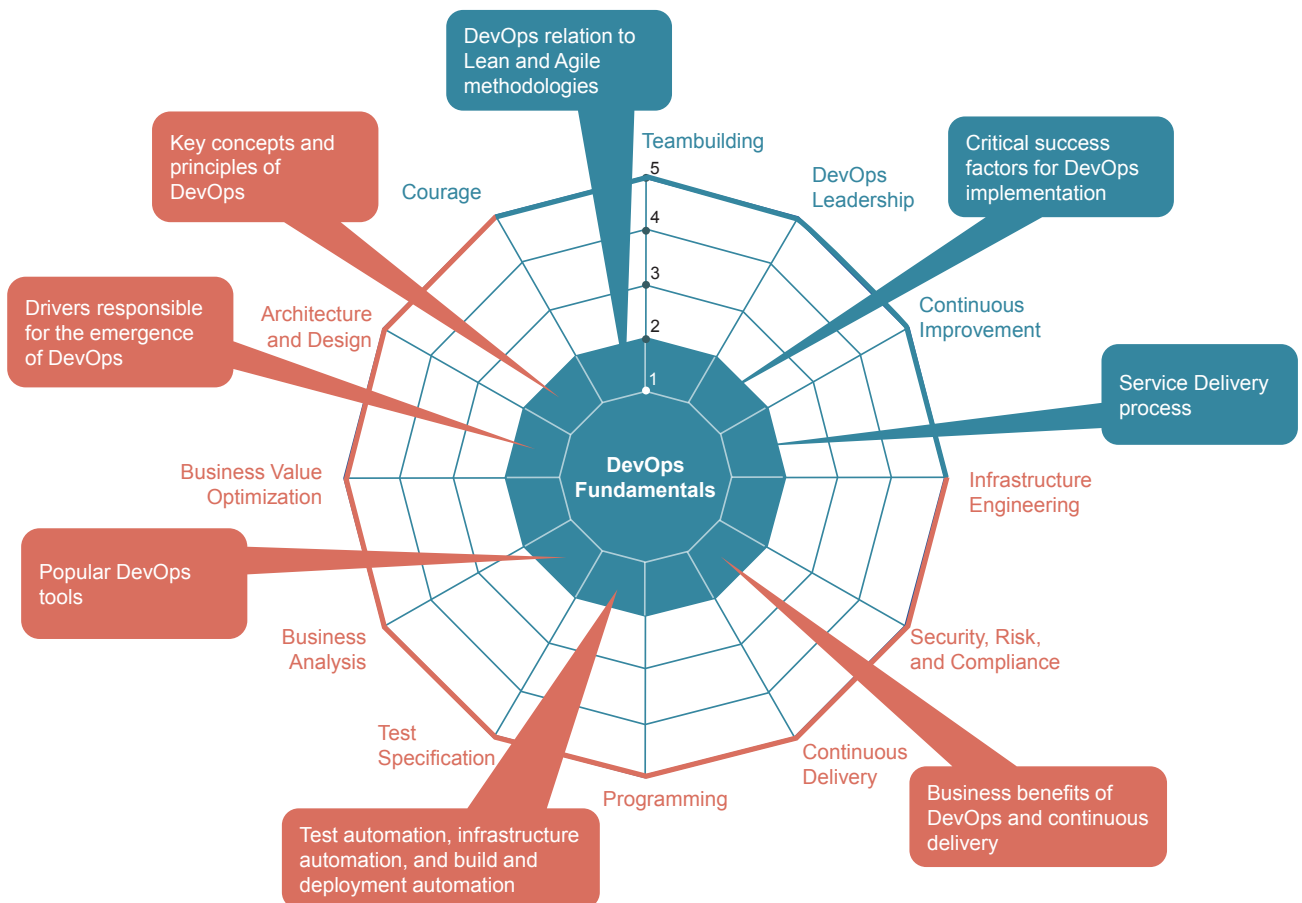
This 3-days course provides learners an extensive introduction to the core Agile DevOps principles. It covers all 12 key knowledge and skill competences that have been defined by the DevOps Agile Skills Association (DASA).



1- Novice / 2- Competent / 3- Proficient / 4 - Expert / 5 - Master

The course objectives and subsequently the module objectives in this course are built around the DASA competence framework. To learn more about the DASA competencies, refer to the “Additional Reading” ebook.

Course Objectives

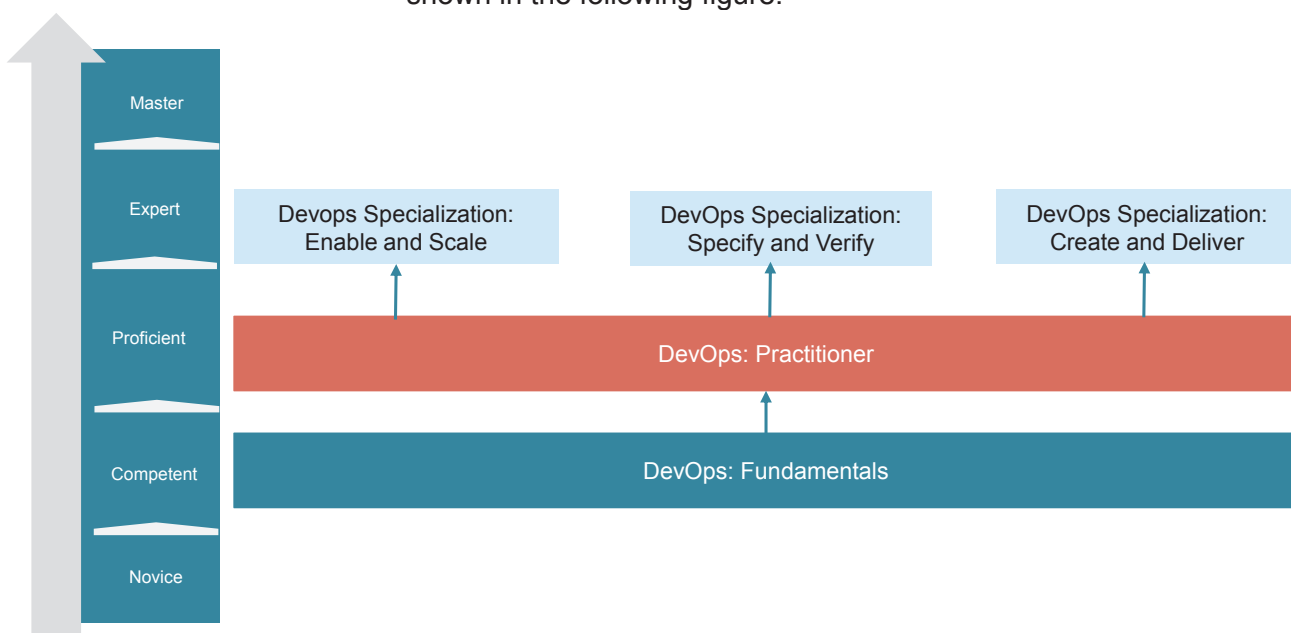


At the end of this course, you will be able to:

- Explain the drivers responsible for the emergence of DevOps.
- Define and discuss the key concepts and principles of DevOps.
- List and explain the business benefits of DevOps and continuous delivery.
- Describe the Service Delivery process.
- Explain the concepts of test automation, infrastructure automation, and build and deployment automation.
- Describe how DevOps relates to Lean and Agile methodologies.
- List the most common and popular DevOps tools.
- Discuss the critical success factors for DevOps implementation.

DASA: Mastering DevOps in Practice (Knowledge and Skills)

DASA Competence Framework recognizes five maturity levels, as shown in the following figure.



Every professional operating in a DevOps team requires all 12 competencies in varying degree. This is described in the DASA professional qualification program.

DASA recognizes five maturity levels ranging from Novice to Master. The intent of DASA certifications is to bring IT professionals from level 1, that is, Novice to level 4, that is Expert.

- The DevOps Fundamentals certification leads to level 2 maturity that is “Competent” on all skill and knowledge areas.
- The DevOps Practitioner training adds level 3 maturity that is “Proficient” on all the skill areas.

- The three specialization modules lead to level 4 maturity, that is, “Expert” on various parts of the competence framework.
- Level 5 is out of scope for DASA and requires much more than just DevOps specific trainings.

Course Agenda

DAY 1	
Module	Subject
01	Course Introduction
02	DevOps Introduction
03	Culture
04	Organization

DAY 2	
Module	Subject
04	Organization (Contd.)
05	Processes
06	Automation

DAY 3	
Module	Subject
06	Automation (Contd.)
07	Measure and Improvement
	Recap
	Mock Exam
	Certification Exam (Optional)

You need to download the Mock Exam from the DASA website, <http://www.devopsagileskills.org/>. In addition, it is not necessary to appear for the Certification Exam on Day 3. You can attempt the exam later as well.

Type of Activities

Group Discussions

The course contains group discussions, spread out in all modules, with the intent of enhancing participants’ understanding, adding context to the content, broadening participants perspective, reinforcing knowledge, and building confidence.

By interacting among themselves and responding to the varying viewpoints, participants tend to learn continually. These discussion

allow the participants to come across the thoughts of their peers, which help them know about each other's past experience, perspectives, and opinions in the context of the topic in discussion.

Exam

At the end of the course, an exam will be conducted. The exam details are:

- **Bloom Level:** 1 and 2
- **Question Type:** Multiple Choice Questions (MCQs)
- **Question Number and Passing Mark:** 40 questions with a minimum passing rate of 65% (26 correct out of 40)
- **Time:** 60 minutes (15 minutes extra for non-native examinees)
- **Exam Type:** Closed book
- **Suggestion:** Recommended that participants take the exam after completion of the course

Course Book

The Course Book is a comprehensive source of information and contains whatever is taught in each module. It is structured in such a way that makes learning easy, simple, and consistent. It also contains several useful exercises related to the course.

The Reference Book can be used as an additional reading material.

Technical Glossary

In order to keep up with terminology, read the glossary available in Course Book. The following table lists some of the technical terminology used throughout the course.

Glossary Terms	Description
Build Process	Process in which software components are packaged into one or more deployment units (packages) that form one or more parts of an application.
Version Control	Also known as revision control or source control, is the management of changes to documents, computer programs, large websites, and other collections of information.
Deployment Process	The process in which one or more deployment units (packages) are moved to a target system and activated for further usage by end user and/or other systems.
Server Provisioning	Server provisioning is a set of actions to prepare a server with appropriate systems, data and software, and make it ready for network operation.
Middleware	Is computer software that provides services to software applications beyond those available from the operating system. It can be described as 'software glue'.
Virtualization	Refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources.

Question and Answer



Activity: *Group Discussion*

Activity Time: 10 mins

Write down your expectations from this training on a sticky and attach it to a wall.

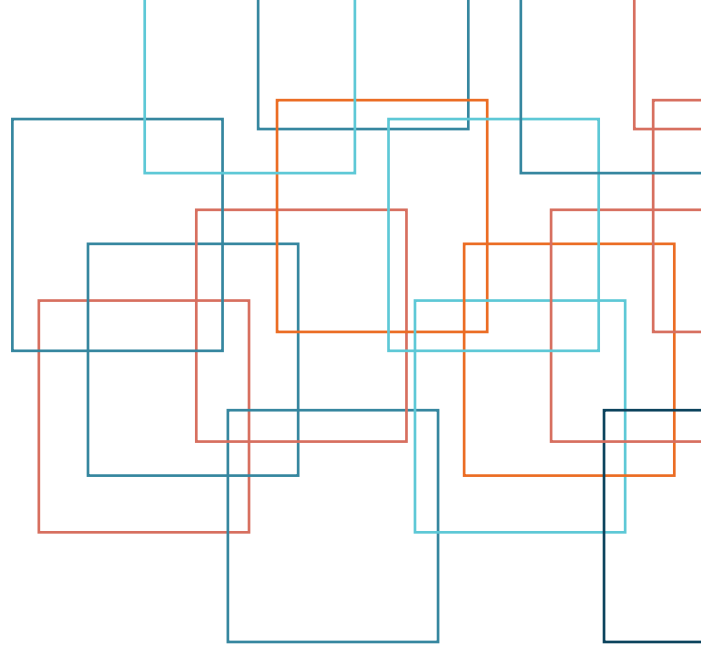
MODULE SUMMARY

In this module, you learned that:

- DevOps Fundamentals is a 3-days course that is designed to provide the basic education required to build your DevOps vocabulary and understand its principles and practices.
- What are the various objectives that this course will help you accomplish?
- What is the 3-days schedule of the training?
- The course contains group discussions and activities for better understanding of the concepts.
- The exam of this course will have 40 MCQs, and its duration will be 60 minutes.
- The Course Book is a comprehensive source of information and contains whatever is taught in each module.

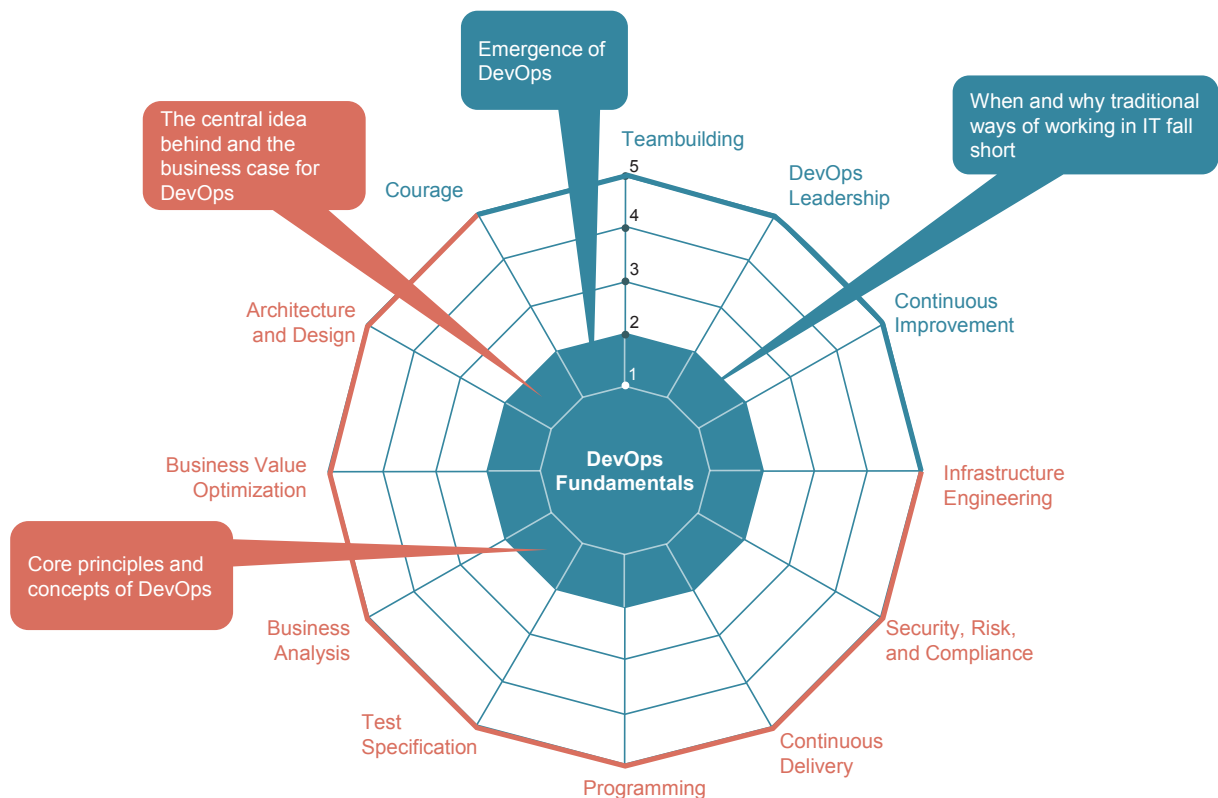
A series of horizontal dotted lines spanning the width of the page, providing a template for writing notes.

2



DEVOPS INTRODUCTION

Module Objectives



At the end of this module, you will be able to:

- Describe the emergence of DevOps.
- Explain when and why traditional ways of working in IT fall short.

- Define the central idea behind and the business case for DevOps.
- Explain the core concepts and principles of DevOps.

Module Topics

- Emergence of DevOps
- Core Concepts of DevOps

EMERGENCE OF DEVOPS

Typical Challenges Traditional IT Organizations Face

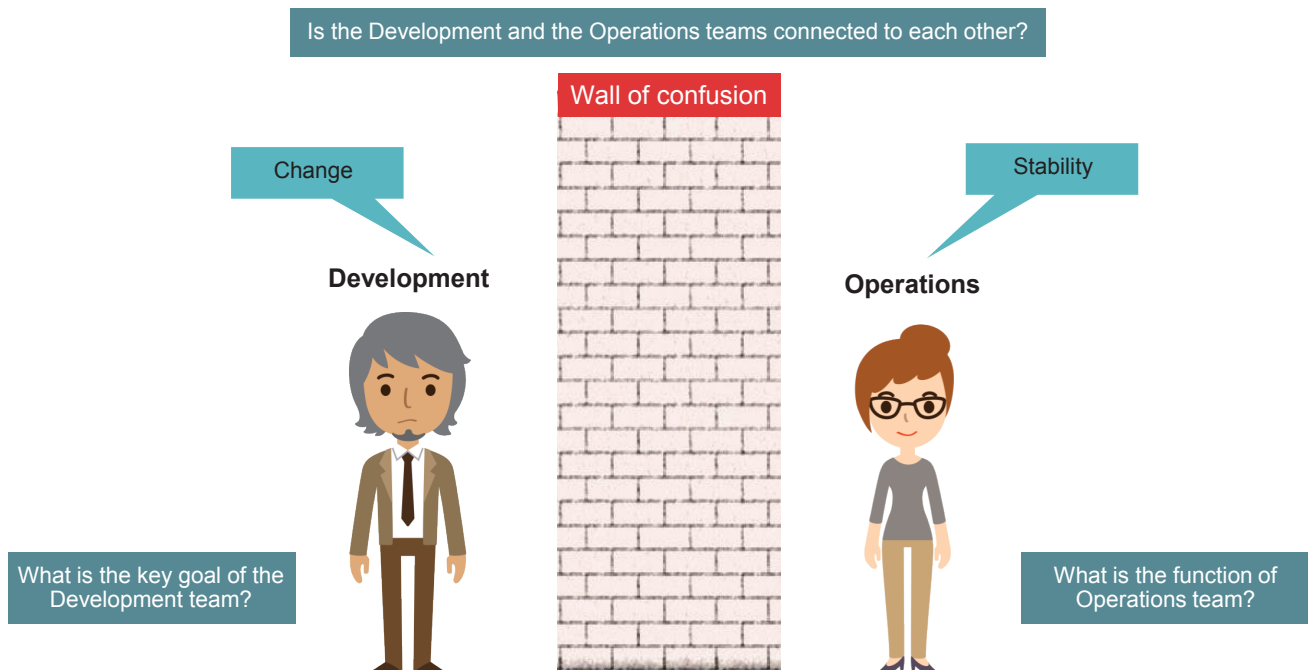


In many organizations, IT continues to be a road blocker than a business enabler. The key challenges faced by organizations are manual releases, low quality of software, sporadic releases, and product backlog resulting in cost escalation. It becomes imperative for the organization to find a solution to these challenges to survive in an extremely competitive and dynamic environment.

The two key teams involved in software development and delivery, Development and Operations, work with contradicting objectives which results in creation of wall of confusion.

Need for Change and Stability Causes Tension

The typical challenge that traditional IT organizations face is to deal with the wall of confusion between Development and Operations.



The **wall of confusion** is combination of conflicting motivations, processes, and tooling. It can result in following problems in production:

- In the absence of a procedural transition from Development to Operations, the Operations team faces challenges in the production. It gets back to Development team for solutions resulting in delay.
- The Operations team is involved in managing Production environment and can share relevant information which can help Development team. However, lack of communication results in loss of sharing.
- The knowledge articles play an important role in building the cohesiveness between Development and Operations team. The knowledge articles include solutions to known challenges and are missed in the environment of conflicting interests between two teams.

Wall of Confusion

Think of the wall of confusion as a solid brick wall where no communication is possible between the people standing on either side. The traditional way of developing software is negatively impacted by this wall of confusion. If the Development team is working on an application, they hand it over to the Operations team only when it is complete. This approach has many loopholes and ultimately results in severe problems in production that causes blast like situation.

Activity: Group Discussion

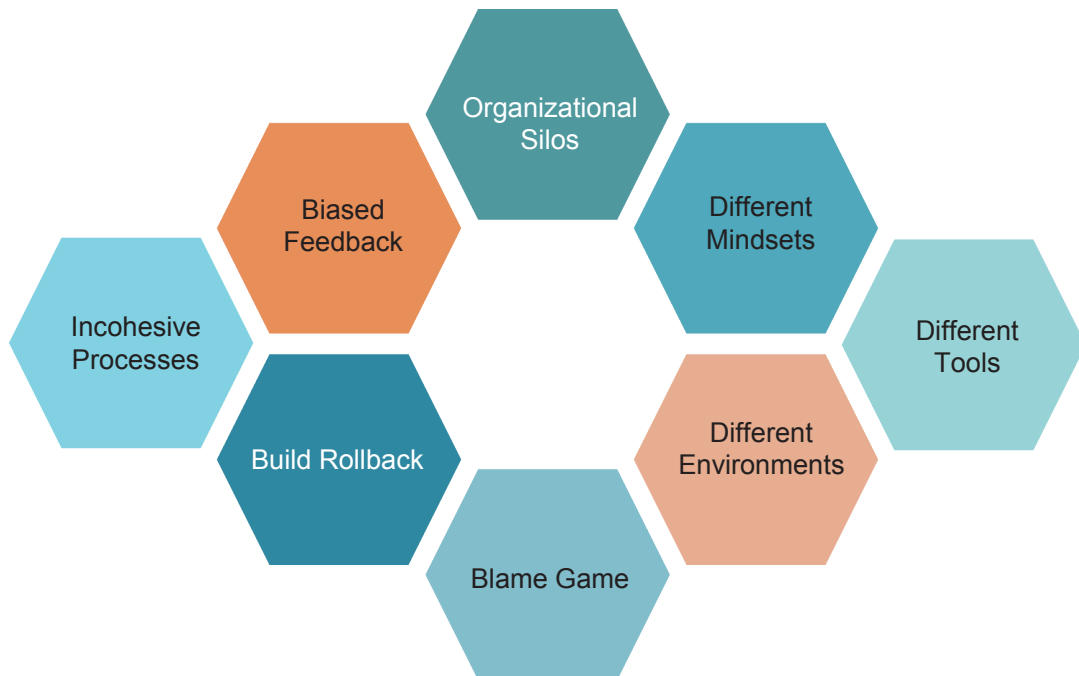
Wall of Confusion

Activity Time: 10 mins

What can be the possible problems that can arise due to the wall of confusion between Development and Operations?

Problems

Although the wall of confusion plays a significant role in the problems IT organizations face, it should be clear that dissolving the wall of confusion involve following challenges:



- **Organizational Silos:** Development and Operations teams work in insolation and interact only during transition. This limited interaction cause unavoidable stress.
- **Different Mindsets:** The Development team attempts to introduce new techniques to bring effectiveness, while Operations team resist these changes as they result in instability.
- **Different Tools:** Development team and Operations team use different tools which cause errors and bugs. Development team uses tools, such as Docker, GitHub. The Operations team uses tools, such as, Jira. Both teams avoid to learn the tools used by other team.
- **Different Environments:** The different environments, such as Development, Test, and Production, are one of the largest sources of errors and bugs raised by the different teams.
- **Blame Game:** Both the teams try to blame each other for the bugs and errors which further escalates the stress between two teams.
- **Build Rollback:** Build rollbacks are frequently required to meet incorrect client requirements, incorrect database in the QA or Production environment, incompatible tools and others.
- **Incohesive Processes:** The processes within two teams are generally based on different frameworks, such as ITIL, ASL, COBIT, and Scrum. The development processes do not

integrate well with operations processes, leading to disjointed processes between both teams.

- **Biased Feedback:** The feedback loops get biased and result in further escalation of confusion.

A Brief History of DevOps

The topic is presented in the class through a video.

DevOps emerged as the answer to the problems caused by the wall of confusion between the Development and Operations. But emergence of DevOps has its own history and series of events. Let us go back in history to analyze the events that gave rise to DevOps.

According to Damon Edwards, a co-founder of DTO Solutions, the DevOps movement was germinated in Belgium back in 2007. Patrick Debois, an IT consultant, was frustrated by conflicting interests and lack of communication between the Development and Operations departments. He found himself straddling between the two teams, while working on a huge data center migration project for the Belgium Government Ministry. Patrick, was, however, confident that there was a better way of working which would allow bridging the substantial gap between the two teams.

At the Agile 2008 Conference in Toronto, Patrick Debois had a meeting with Andrew Shafer, a partner at Reductive Labs. They both had in-depth discussion about their mutual frustrations. This discussion gave rise to the Agile System Administration group on Google Groups by Andrew Shafer and Patrick Debois. Although the group was not overly popular, it leads to some fascinating discussion.

Moving forward in the year 2009, John Allspaw and Paul Hammond of Flickr made the famous presentation “10 Deploys A Day” in Velocity Conference at San Jose. In the same year, DevOpsDays was born, and the first event was conducted from October 30th to 31st at Ghent in Belgium. The event attracted administrators, developers, and managers from all around the world. The event’s success inspired other DevOpsDays events in different countries. The DevOpsDays events acted like a catalyst for the conversation and a grassroots movement.

During the same time, the IT industry was flourished with converging adjacent technologies such as Agile, Lean, Operations Management, and IT Service Management. This resulted into series of debates and discussions at conferences and twitter throughout 2009 and 2010. These discussions eventually formulated underpinning philosophy that has become DevOps. This convergence is popularly referred to as “Perfect Storm”.

While the DevOps Practitioner such as Patrick Debois and Andrew Shafer had a strong desire to popularize the DevOps movement, the movement was not gaining much attention from vendors, analysts and large scale enterprises. To ignite the movement, the ongoing DevOps

FOOD FOR THOUGHT

Learn about the different frameworks, such as ITIL and COBIT, and try to understand the relevance of processes in these frameworks.

community and collaboration began to analyze various tools used for development and operations and in 2011 start to build open source tools such as Vagrant. A tidal wave of agile and open source toolsets including tools such as Chef, Puppet, Jenkins, Hudson, JuJu washed away the old legacy toolsets.

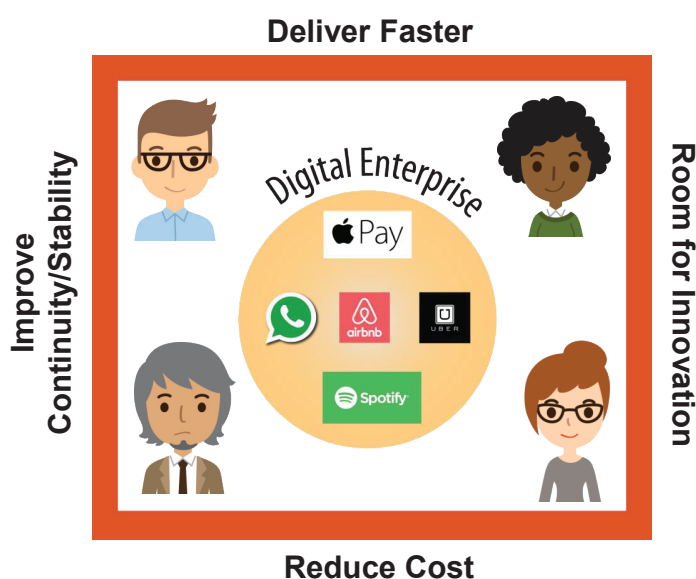
With the emergence of enterprise class open source tools and initiatives such as the DevOpsDays, DevOps started moving towards Enterprise scale adoption across leading global organizations. By the year 2012, DevOpsDays were popping up around the world and become the must-attend events to know about the latest and innovative thinking and happenings in the DevOps world.

The next year 2013 was the year of rise of DevOps texts. A flood of DevOps related books such as the Phoenix Project by Gene Kim were released. One important voice in the DevOps texts is Mike Loukides, vice president of content strategy for O'Reilly Media. He, along with Debois, edits some of the most fundamental DevOps texts.

During the year 2014 and afterwards, DevOps is crossing into the enterprises and established brands such as IBM, Target, and Nordstrom embrace the DevOps movement. But this period is also witnessing the transformation in DevOps philosophy. DevOps is getting more mainstream in organizations and moves out of the pure Development and Operations space. The emphasis is put on creating flow in organizations and creating a culture of high-performance IT. This means that DevOps does not purely focus on automation alone, or emphasizes methods, frameworks and products in the IT space, but covers organizational transformation, culture, way of working and people aspects as well.

Reference Reading:

<https://blog.newrelic.com/2014/05/16/devops-name/>



Benefits of DevOps

Fast movers displace traditional companies in all industry domains. To survive, companies need to radically rethink their IT strategy. Where will your company be in five years?

What is commonality?

- IT is the strategic differentiator.
- Fast movers use Automation, Continuous Improvement, and simplified operating models.
- Operations and Development are in sync.

Software lends itself very well for fast and dynamic delivery!

The figure sums up the most compelling business benefits of DevOps. DevOps ensures a faster time to deliver the product, reduction in production cost, room for innovative products/ideas, and improvement in stability.

These key benefits are driving organizations to break down the wall between business and IT. The organizations have started to realize that IT offers a strategic differentiation. Embracing a digital transformation is now key to survival and the four key goals listed on the figure are now placed high on the agenda of top management.



Marc Andreessen said: "Software is eating the world". The pressure to deliver faster, better and cheaper software products has considerably increased the focus on DevOps principles.

Antifragility

In order to stay in business, digital enterprises need to be antifragile organizations.

Antifragility is the ability of systems (or organizations) to get better as a result of shock, disruptions or disorder.

THE ANTIFRAGILE ORGANIZATION		
MANAGEMENT INNOVATION Multiply funding for new initiatives Learn from the fringe Community over hierarchy Employees first Ensure transparency in decision making Kill bureaucracy Think competencies and platforms Honor Web-inspired value Reinvent management Speed	LEAN STARTUP Implement experimentation systems Experiment is a product Instill entrepreneurship Customer first Validated learning Minimum Viable Bureaucracy (MVB)/lean Think digital innovation Honor end customer value Self-management Pivot	DEVOPS Continuous iterative innovation Embrace a culture of 'Fail Fast' Make DevOps teams responsible Empower employees to put customers first Short feedback loops & measure end-to-end System thinking, crossing silos Innovate digitally and use cloud platforms Honor end customer value Self-management Flow
Human component back into focus	Stop wasting people's time	Accelerate innovation

Source: *Design to Disrupt Whitepaper, Mastering Digital Disruption with DevOps, Sogeti VINT, March 2016*

When the concept of antifragile is applied to an organization, it implies creating an environment what is adaptable to stressful and unexpected atmosphere, leading to a better transformation. Management Innovation, Lean Startup, and DevOps are three integral characteristics of an antifragile organization. DevOps aims to be the cultural aspect. Lean Startup tends to focus more on a method for product development. Both of these explicitly work on Management Innovation and put up a vigorous fight against bureaucracy, make teams and staff responsible, and urge the customer to get involved when it comes to digital innovations. This is how speed, staff engagement, and customer obsession come within reach of any organization.

FOOD FOR THOUGHT

Are "antifragile organizations" same as "robust organizations"? Think about it!

IT organizations need to be antifragile as the world of IT is changing rapidly. Businesses need their IT organizations to be responsive and need their IT architectures to be adaptable to shocks or disruptions. DevOps attempts to create this kind of environment; creating organizational units that can autonomously act on the software and hardware required to deliver the IT service. DevOps requires short delivery times to ensure disruptions (in the form of different choices resulting from market changes) can readily be accommodated and used to improve the IT service.

Business Case: The High-Performing IT Organization



TIPS

The current State of DevOps report can be downloaded from the following link: goo.gl/tQWP3q.

Source: *State of Devops report 2017*

The preceding numbers are taken from the State of DevOps 2016 report. Over the past 5 years, they have surveyed more than 25,000 professionals worldwide to better understand how DevOps practices impact IT and organizational performance. You can refer to the latest State of DevOps Report for more details.

There is a tangible difference between high-performing organizations and their low-performing counterpart. One of the reasons that enable high-performing organizations to perform better is their style of working according to a set of patterns. This set of patterns is known as DevOps.

Business Case: Seven Reasons for DevOps



Moving forward with regard to the points stated in the “State of DevOps report”, let us analyze the seven big reasons for adopting DevOps.

1. **Improved speed to market:** DevOps approach enable an organization to go from an initial concept to a viable product in a shorter timescale than was traditionally acceptable with a Waterfall approach.
2. **Continuous Integration and delivery:** Continuous integration is a practice that encourages developers to integrate their code into a main branch of a shared repository early and often. DevOps approach and culture allows organizations to deploy code more frequently, with shorter lead times.

Functional and Non-functional Requirements

Functional requirements are associated with the tasks a product/service should do. Non-functional requirements refer to the attributes of a product/service, such as autonomy, resiliency, maintainability, testability, scalability, and reliability.

3. Higher quality, fewer failures, and higher stability:

Higher quality: Adoption of DevOps allows systems to meet the explicit **functional requirements** and implicit **non-functional requirements** leading to creation of high quality products.

Fewer Failures: The 2015 State of DevOps report highlighted a trend that the organizations who adopt a DevOps mindset and culture have 60 times fewer failures than those not implementing a DevOps approach.

Higher Stability: DevOps allows an individual group to take charge of development and stability of systems. Deploying in small numbers and frequently allows a faster troubleshooting. Faster troubleshooting allows increased stability.

4. Innovation and creativity:

Automated deployments, continuous integration, and standardization allows people to focus on the ingenious aspect of their work.

5. Increased employee engagement and job satisfaction:

DevOps fosters a multi-skilled environment which creates a learning platform. Learning new skills and their usage enhances employee engagement.

6. Breaking down silos and eliminating waste; It is all about collaboration!

DevOps creates cross-functional teams which discourage silos in the organization. DevOps nurtures collaborative spirit in the organizations. In complex enterprises with many fast-paced IT innovations, investing in collaboration helps to leverage all knowledge and skills available.

7. Resource and cost reduction:

Adoption of DevOps inculcates an environment which supports continuous delivery and Lean Management practices. Short cycle times and high quality reduces costs effectively and resources are optimized.

RESULT: Increased Performance

Standardized production environments and automation tools help make deployments predictable. These processes free people from routine tasks, allowing them to concentrate on the more creative aspects of their role. Consequently, it leads to increased performance of the people.

CORE CONCEPTS OF DEVOPS

Some DevOps Definitions

“DevOps isn’t a thing. It’s not a product, standard, specification, framework or job title. DevOps is about experiences, ideas and culture. It’s about the close communication and collaboration between IT operations and development, and how they can improve the products and services that they produce by thinking differently about how they work together, using a new mentality.”

Gareth Daine, Devops Evangelist

“Fundamentally, DevOps is the activity of optimizing the development-to-operations value stream by creating an increasingly smooth, fast flow of application changes from development into operations, with little waste. Optimization of the value stream takes place continuously using various continuous improvement techniques like the Toyota Kata.”

Dave Roberts, Executive Advisor at BMC Software

The definitions listed here highlight the following elements of DevOps:

- DevOps is a culture that nurtures collaboration.
- DevOps allows an environment in which the mutual understanding about each other roles and responsibilities exist.
- DevOps is a movement inspired by practitioners for practitioners.
- DevOps allows pursue of common goal aimed at high quality and cost effective products marked with fewer failures and bottlenecks.

These elements stand for something different from and larger than a ‘thing’, such as a product, standard, or framework. This makes DevOps intangible but also applicable to enterprise-wide IT improvement and continuous innovation of the IT capability.

DevOps is the Culture of High-Performance IT

DevOps is a **CULTURAL** and **OPERATIONAL** model that fosters **COLLABORATION** to **ENABLE** high-performance IT to **ACHIEVE** business goals.

Adopting the DevOps practices help an organization to create a culture of high-performance IT. The cultural and operational model underlying the core vision and principles of DevOps is the key to arriving at this high-performance level. The focus on culture means that people in an organization need to drive to continuously improve the state of the practice, together with colleagues and peers.

DevOps is Tightly Intertwined with Agile and Lean IT

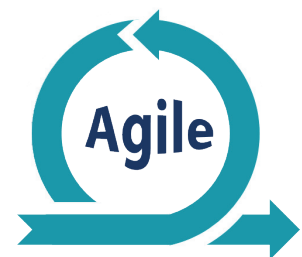
The underlying principles of Agile and Lean IT are at the core of DevOps and the three are tightly intertwined. Therefore, DevOps practitioners often choose to use many of the same techniques as found in Agile and Lean IT.

Reference Reading:

<http://theagileadmin.com/what-is-devops/>

FOOD FOR THOUGHT

Find out some other definitions of DevOps and ponder over the similarities and difference between these definitions.



Agile Methodologies

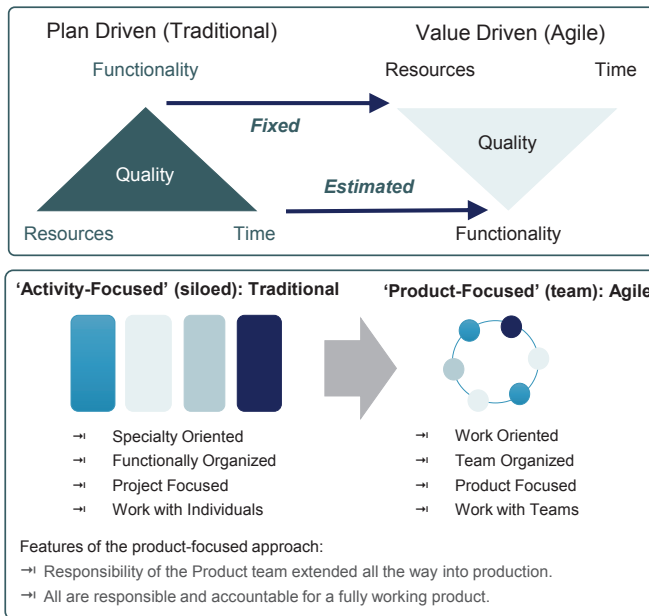


Lean IT

DevOps finds many of its practices deeply rooted in Agile and Lean IT organizations. Lean IT and Agile both originated from the application of Lean principles to an IT organization (in the case of Lean IT) or software development (in the case of Agile). DevOps makes good use of Lean IT and Agile practices and methods that have already proven to be successful and that take the delivery of IT services to the next level.

Agile: Satisfy the Customer

Agile movement provides alternative to traditional project development.



Traditional Development

- Start with a complete design
- Building is followed by testing the final product
- Finally, testing in practice
- No feedback loops
- Plan driven

Agile Development

- Every Sprint delivers working software that can be used in practice
- Starts with delivering basic functionality to which features are added
- Value driven

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

Reference Reading:

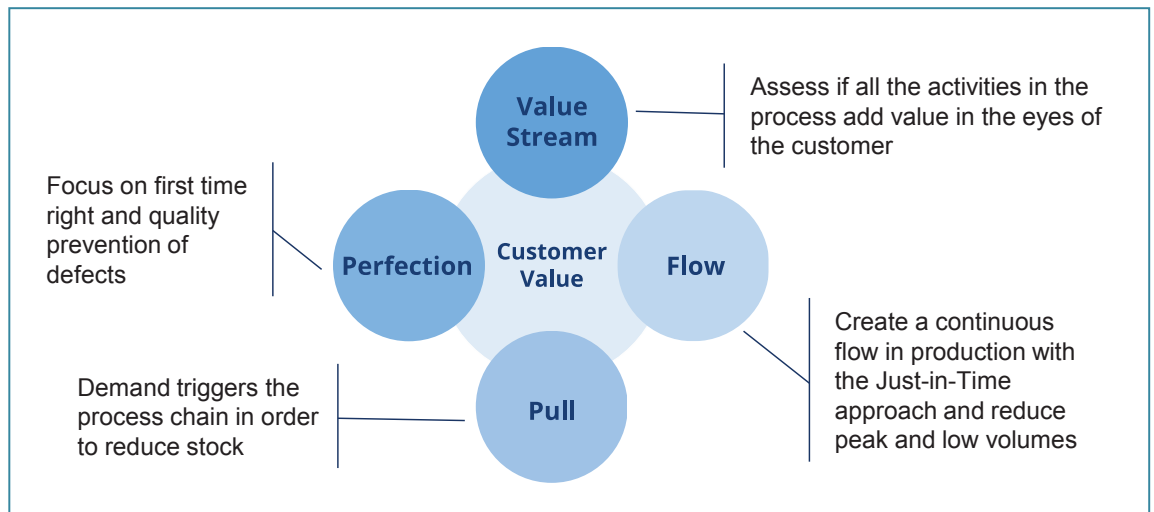
“The Art of Agile Development” by James Shore

Agile refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile breaks the development of new functionalities into smaller functional units according to user stories. The units are continuously prioritized to deliver the software in short cycles known as iterations.

DevOps in many ways fits uniquely well with Agile. The two unique approaches are almost synonymous except for prevue in scope. The convergence of development methodologies and delivery solutions has paved the way for numerous companies to reap huge success and improvements.

Lean: Customer Value at the Center

Lean focuses on creating the value for customer.



Lean is delivering value to customers and continuously improving the ability to do this, by removing waste from the entire system that produces the value.

Lean principles took birth in Toyota production factory with the aim of minimizing the resources. Lean principles provide the foundation for building the DevOps culture. For example, Value Stream Mapping, a tool used in Lean for removing the imperfections out of the process chain, finds a ready acceptance to visualize the DevOps value stream and measure it. Such measurements help improve the delivery pipeline by eliminating bottlenecks, and making it more efficient and productive. Organizations can apply Lean principles in many contexts, tools, and methods with multiple sources.

FOOD FOR THOUGHT

Learn about the reasons that led to the need of Lean adoption at Toyota.



TIPS

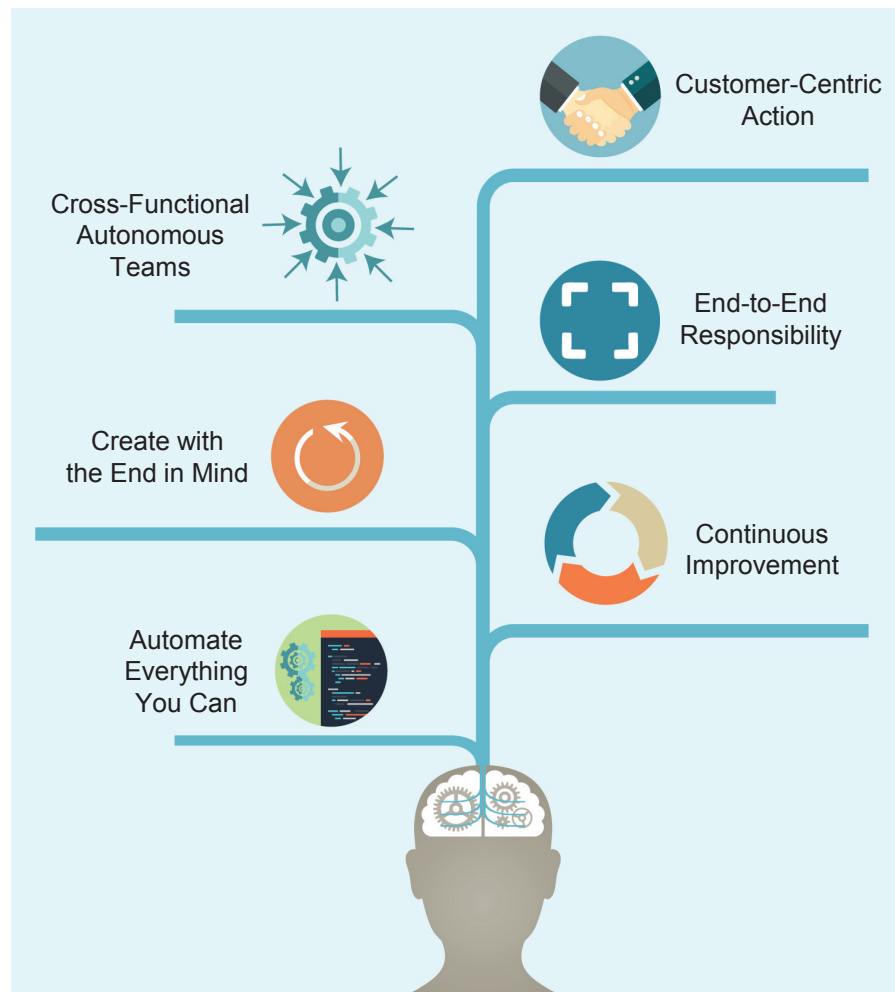
Value Stream mapping is a method for analyzing the current state and designing a future state for the series of events that take a product/service from its beginning through to the customer. At Toyota, it is known as “material and information flow mapping”.

Activity: Group Discussion

Activity Time: 10 mins

What do you think are the core principles when “going DevOps”?

DevOps Core Principles



Reference Reading:

DASA Whitepaper - Embracing Digital Disruption by Adopting DevOps Practices (goo.gl/LfUZX6)

Many of the definitions of DevOps adequately explain one or more aspects important for finding flow in the delivery of IT services. Instead of trying to create a comprehensive definition on your own, refer to the following six principles when adopting or migrating to a DevOps way of working:

1. **Customer-centric Action:** All activities involved in building IT products and services should revolve around customers. Short feedback loops with customers and end-users help to understand the changing requirements.
2. **Create with the End in Mind:** This principle focuses on working towards creating product and services which meets the needs of customer. This principle considers taking a holistic view of both the creation and use of the IT product/service.
3. **End-to-End Responsibility:** End-to- End responsibility implies taking full accountability for the quantity and quality of a product. In a DevOps organization, teams are vertically organized so

that they can be fully accountable for the products and services they deliver.

4. **Cross-Functional Autonomous Teams:** In product organizations with vertical, fully responsible teams, the teams need to be fully autonomous throughout the entire lifecycle of the product.
5. **Continuous Improvement:** In DevOps culture, strong focus is put on continuous improvement. Continuous improvement is achieved through minimizing waste and optimizing speed, costs, and ease of delivery.
6. **Automate Everything You Can:** Automation reduces errors, improves accuracy, reduces delays and removes tedium.

DevOps Principle #1: Customer-Centric Action

DevOps encourages an open culture that has the following characteristics:

- No bureaucracy
- No fear of asking questions
- Risk taking
- Innovating

Courage

Innovate

In an open culture, teams are open to feedback. There is no hesitation or barrier to ask questions and calculated risk taking is encouraged.

Adopting a customer-centric action is a key principle behind DevOps adoption. Creating a positive customer experience adds value to the organization. Short feedback loops with real customers and end-users ensure a customer-centric action. In a DevOps environment, teams and individual have both the courage and the authority to act to fulfil customers' needs.

DevOps Principle #2: Create with the End in Mind

If you do not know where you are going, you will not know whether you lost the way. Create with the end in mind brings focus on the end results. This will foster the product, service thinking, and collaboration, which is one of the key ingredients to DevOps. However, it requires an engineering mindset and mutual trust among different teams and team members.

This DevOps principle insist on moving away from process-oriented models to models that encourage units or individuals to work on a particular area rather than oversee the complete picture. This is the Agile way of working that delivers single changes or in small batches to customers so that potential issues can be identified immediately. The key is to get feedback quickly about what their customers find most valuable.



TIPS

To meet customers' requirements, DevOps organizations need to act as Lean startups that:

- Innovate continuously.
- Adjust when a certain strategy is not working.
- Constantly invest in products and services that receive a maximum level of customer delight.
- Rapidly respond to changing or emerging customer needs.

Product and Service Thinking

Engineering Mindset

Collaborate

Live Your Accountability

Concept to Grave

Performance Support

**TIPS**

Think about how an organization can flourish by establishing cross-functional teams in contrast to competence-based teams or a culture of heroes.

DevOps teams need to act as (Lean) “product companies” that explicitly focus on building working products sold to real customers. Such teams adopt Lean thinking to focus on creating flow and become experts in problem solving (the Lean Kaizen mindset).

DevOps Principle #3: End-to-End Responsibility

Caring about the end-to-end responsibility might be the most crucial ingredient for DevOps. When people care and have the required skills, knowledge, and resources, they can and will collaborate to live up to their responsibility. If they care, they will learn, adapt, improve, and provide great services and value.

In a DevOps organization, teams are vertically organized so that they can be fully accountable for their services. The teams are kept stable to ensure effective working habits can be embedded. These teams provide performance support to products until these reach end-of-life. Such an end-to-end support enhances the level of responsibility and the quality of the engineered products.

DevOps Principle #4: Cross-Functional Autonomous Teams

Cross-functional teams consist of representatives from all disciplines responsible for developing and deploying an IT service. These teams are fully empowered and self-sufficient to design, build, test, deploy, and run the software. To be able to do this, a team needs a balanced set of skills and team members with T-shaped profile and complementary skills.

T-Shaped Profiles

Complementary Skills

“There’s no such thing as a DevOps Hero”

DevOps is synergistic. It requires people to collaborate effectively by:

- Ensuring they have overlapping skills and knowledge, combined with complementary specialist skills and knowledge (T-shaped profiles).
- Giving feedback to each other.
- Avoiding blame evaluations (no blame game or finger pointing).
- Trusting each other (Having a high-trust culture has a strong impact on both IT performance and organizational performance.).

DevOps Principle #5: Continuous Improvement

Continuous improvement is an approach to identify opportunities to streamline work and reduce waste.

If it Hurts, Do it more Often

Fail Fast

Experiment

"What cannot be measured, cannot be improved"

Continuous improvement is a dominant concept borrowed from the Lean movement that focuses on adaptability and learning through structured problem-solving. It requires teams to focus on experimentation, minimizing waste, optimizing for speed, cost, and ease of delivery to continuously innovate. It is, therefore, important to experiment and learn from failures (try to fail as fast as possible).

To continually improve a product/service, it needs to be measured. A DevOps implementation should be designed to measure everything, such as processes, people, and tools.

DevOps Principle #6: Automate Everything You Can

Regardless of the technology platform or development practices, every organization uses a process to build new software and IT services. This process can be manual or automated. Moving away from manual efforts to automation derives efficiency and consistency in the process. Automation helps to:

Enhance Quality

Maximize Flow

There are several ways that can help increase the speed, reduce the cost, and enhance the quality of IT, such as:

- Continuous improvement (waste reduction) is greatly helped by automation.
- Continuous delivery focuses on bringing software into production through a fully automated process, multiple times per day without problems.

Measurement Through Metrics

DevOps seeks measurement of processes, people, and tools through following metrics:

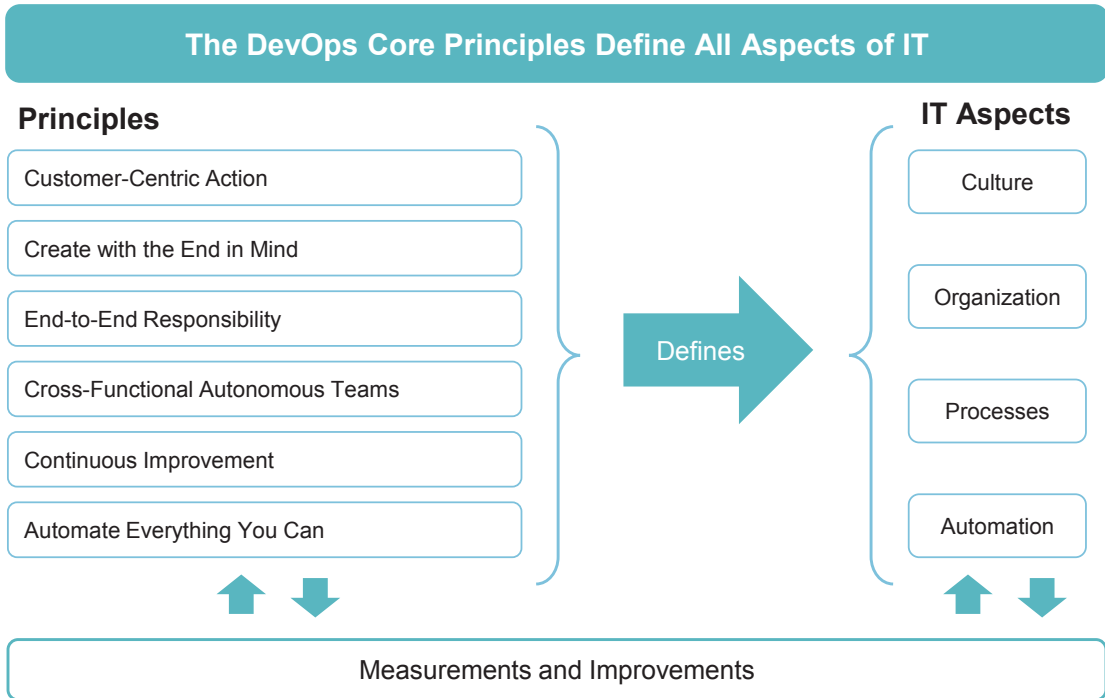
- Performance Metrics
- Process Metrics
- Innovation Metrics
- Culture Metrics
- Support Metrics

FOOD FOR THOUGHT

Think about how automation leads to enhance quality and maximize the flow of value to customers.

- Cloud-native platforms replace traditional data centers.
- Container-based infrastructure help treating Infrastructure as Code (IaC).

DevOps Principles and Aspects of IT



The core DevOps principles impact the various aspects of IT, such as culture, organization, processes, and automation. You can use the DevOps principles to select, implement, and evaluate best practices for IT (culture, organization, processes, and automation) optimization. In addition, you can perform measurements and improvements to implement a continuous feedback or continuous improvement loop. For example, “Automate Everything You Can” need a culture and mindset that promotes automation and optimization of the organization, processes, and technology used.

Activity: Group Discussion

Activity Time: 15 mins

One of the principles of DevOps is to automate everything you can. How much of the delivery of IT services through the entire lifecycle is automated in your organization?