DASA DevOps Fundamentals

Pre-Course Reading

DASA DevOps Fundamentals

Copyright and Disclaimer

DASA DevOps Fundamentals | r2.0.1

Copyright

Trade Marks

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 publisher is also not responsible for the discontinued service if any, for example, the Web links in the course might not work in future.

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, the publisher accepts no responsibility for any loss or inconvenience caused due to the regional differences in the usage of the English language.

Contents

My Notes

The purpose of the pre-course reading material is to enable you to know about:

- What is DevOps?
- How is it originated?
- What is DASA?
- What are the various DevOps practices?

The knowledge will enable you to understand the concepts in the classroom and motivate you to participate actively in group discussions.

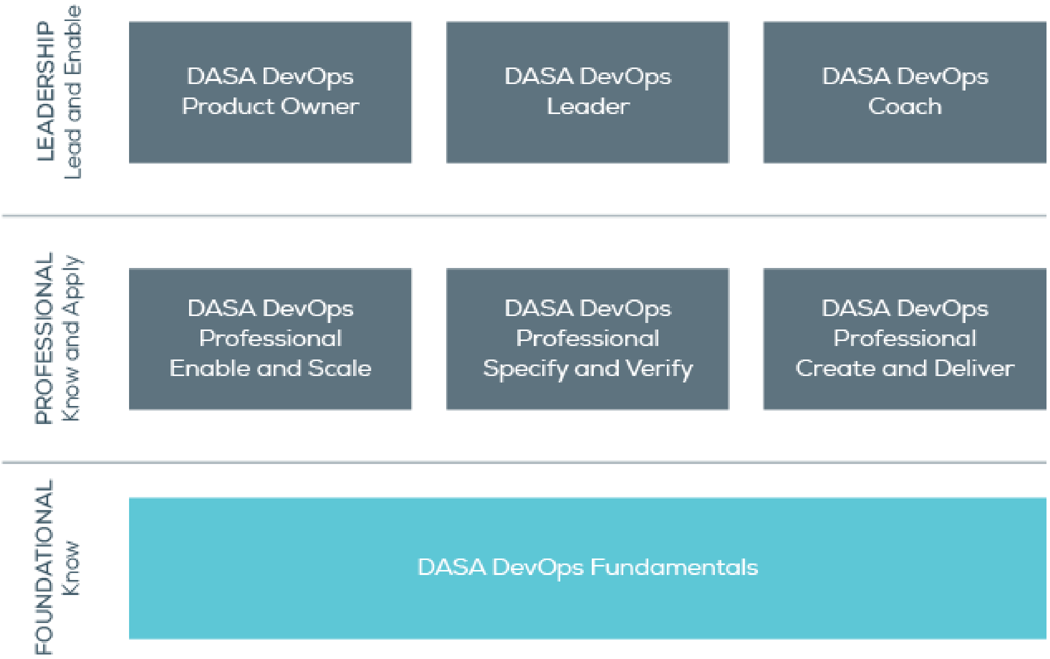
DASA DevOps Fundamentals

Up to 200 times faster software deployment, 30 times increased deployment frequency, and 60 times higher change success rates, organizations such as Netflix, Spotify, and Facebook are revolutionizing the IT game by successfully implementing DevOps principles. The data does not lie. You do not have to be a hot Web company or a monster enterprise to be a DevOps leader. Companies, large or small, and young or old, have magnificently made the transition and have the proof of success in their pockets.

DevOps training is the starting point for an organization going on the DevOps journey. Improved workflows and faster deployments starts with a core understanding of DevOps fundamental concepts by anyone involved in an Agile and/or DevOps team.

DASA develops and evangelizes a vendor-neutral DevOps qualification program for professionals, generates interest and awareness for the need for knowledge and skill development, promotes open source certification for DevOps knowledge and skills, and ensures the quality of training for the market through a logical and threshold-driven qualification program.

Anyone can participate in defining role-based competencies, learning paths, and qualification schemes. All existing learning content that maps against the DASA knowledge and skill areas has value. DASA will map content and demonstrate relevance and will maintain an open and logical operating model for training delivery, as shown in the following figure.

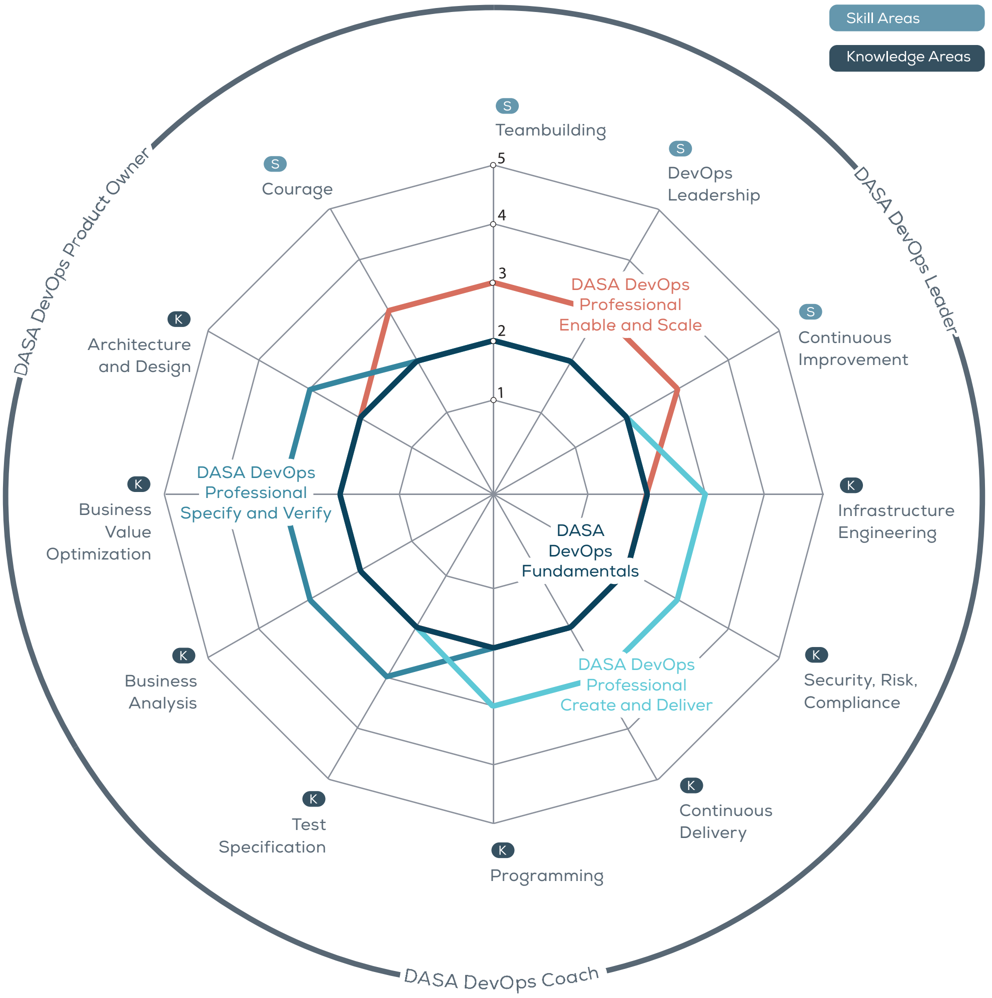

LEADERSHIP
Lead and Enable
DASA DevOps Product Owner
DASA DevOps Leader
DASA DevOps Coach
PROFESSIONAL
Know and Apply
DASA DevOps Professional Enable and Scale
DASA DevOps Professional Specify and Verify
DASA DevOps Professional Create and Deliver
FOUNDATIONAL
Know
DASA DevOps Fundamentals

DASA offers certifications to all key profiles and comes up with the certification program, as shown in the preceding figure. It includes the following three levels, ensuring the right certification for the right audience:

- The **Foundational** level focuses on 'knowing' and helps individuals to build an understanding of DevOps, its principles, DASA's approach to DevOps, and puts DevOps into a business perspective.
- The **Professional** level certification level provides capability oriented certifications. The certification helps professionals learn the key traits of their job and how to apply DevOps in real life. This level includes three certification programs, one for each of the professional profiles that we identify in a team.
- The **Leadership** level focuses on the ability to lead and enable. This program does not focus on building capabilities. It also does not explain the tools that are there in the toolbox for the professional. It focuses on how the professionals can best operate in his/her role, such as managing the process, removing barriers for people, and leading the team.

The DASA DevOps Competence Model

The DevOps Agile Skills Association (DASA) competence framework identifies 8 knowledge areas and 4 skills that are relevant in DevOps, as shown in the following figure.



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

Every individual operating in a DevOps team requires to be competent at all 8 knowledge areas and proficient at the 4 skill levels. In order for DevOps teams to be effective, they require all 12 areas to be at the Expert level. Individual team members can specialize in specific areas, in order for teams to achieve these capabilities

Introduction to DevOps

Fast-paced changing requirements is one of the most notable characteristics of these days. Consequently, we need a fast response to these new business requirements, but there are still ITs with piled-up delayed requests, low-quality outputs, infrequent releases, and other daily problems. In many organizations, we can even see that IT is not a business enabler while it is more of a bottleneck. These are some typical challenges that traditional ITs are facing, and they need a way to deal with them.

Looking through traditional IT organizations showed us that working in a siloed structure alongside the existence of "wall of confusion" can be considered as one of the most significant causes of this situation. Wall of confusion is a physical or logical wall that separates teams like Application Development and Operations from each other. When teams do not have real communication and collaboration in place, then they won't understand each other's needs, priorities, vision and they won't support each other for delivering a quality service or product in short lead-time. Here is the beginning of blaming games and teams accuse each other of the poor results.

On the other hand, there are companies displacing traditional companies in all industry domains very fast. These companies which are also known as "fast movers" started to overcome those problems. They started breaking down the wall between teams and encourage them to work alongside each other. Fast movers like Uber, Spotify, WhatsApp, Airbnb, etc. have something in common:

- IT is playing a key role in the company and used as a strategic differentiator in the market
- Keep distance from complex operating models and focused on more simplified ones
- Use automation at the highest possible levels
- Cherish continuous improvement culture
- Operations and Development work together and in sync

Emergence of DevOps

Like many other frameworks and methodologies, DevOps is also showed up as a solution to a problem. Back in 2007 a Belgian consultant, project manager, and agile practitioner Patrick Debois took on an assignment with a Belgian government ministry to help with data center migrations. In that project, he needed lots of communication and collaborations between application development teams and the operation teams (server, database, network). It is so apparent that he faced real challenges for managing what he needed and these challenges led him to find a better way.

In 2008, he met with Andrew Schafer at the Agile Conference in Toronto.

They discussed their concerns and decided to share what they think with others and receive feedback using a Google Group named "Agile Systems Administrator." This idea was not that successful and did not gain much attention.

The year 2009 and the O'Reilly Velocity Conference was an accelerator to what started. In that conference John Allspaw, senior vice president of technical operations, and Paul Hammond, director of engineering from Flickr gave a now-famous presentation titled, "*10+ Deploys per Day: Dev and Ops Cooperation at Flickr.*" This presentation highlighted the common challenges between application development teams and the operation teams and also their root causes. Their presentation made it clear that the only way to overcome such a situation is changing the way in which application development and operations work and behave. They highlighted that changing organizational culture is a must and makes teams' activities transparent and fully integrated for such a transformation.

Debois was unable to attend this event and watched it using a video stream. He was inspired and soon formed his own conference called DevOpsDays in Ghent. It was a time where the term "DevOps" officially used for the first time.

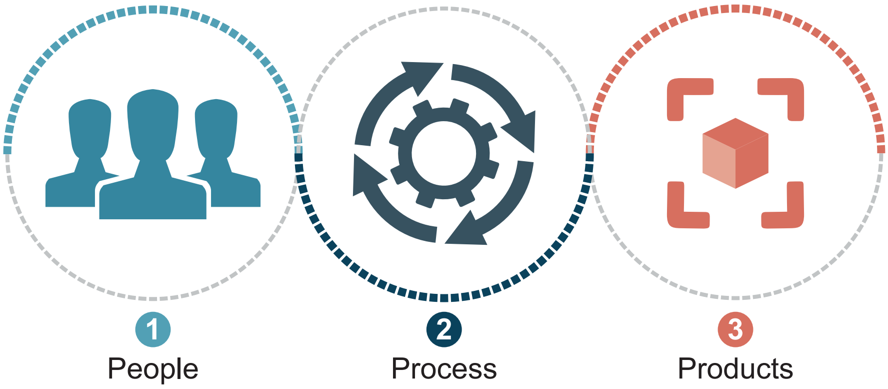
DevOpsDays conference is held for the first time in the United States in Mountain View, California, back in the year 2010. Now you can hear about DevOps ideas and presentation in most events and conferences.

The year 2013 was the starting point to the rise of DevOps publications. "The Phoenix Project," written by Gene Kim, Kevin Behr, and George Spafford, was one of the most famous books in this context. This book is a fictional novel tells the story of an IT manager who assigned to manage a mission-critical e-commerce development project which is not in a good situation. This guy was introduced to disciplines of lean manufacturing and ended up with the concepts of DevOps. In those years, many other books and publications related to DevOps released to the market.

It is now more than 10 years that DevOps exists and in all these years, supporting tools such as Vagrant, Puppet, Chef, etc. played a crucial role in trending this concept in the market.

DevOps and Dimensions

Back in the year 1964, Harold Leavitt's published a paper titled "Applied Organization Change in Industry." In this paper, he presented a model that can be considered as an origin for the "People, Process, and Products/Technology" phrase. Harold's article and also an idea behind this phrase both consists on this crucial factor that if you want to change something in your organization you need to have a specific plan and budget for all these three (People, Process, and Technology). You can not ignore one and pay more attention to one another.



Let's consider fast movers as our context. What is your imagination for the people side which includes organizational structure, organizational culture, staff skills, and their competences? What about the processes? And also tools and technologies? Your response can be imaginable. As a predictable answer, everyone expects a simple organizational structure, healthy corporate culture with happy multidimensional skilled staff. Also, having simplified and optimized processes in-place with the highest level of automation.

Moving toward this state will be an organizational transformation journey which needs a clear transformation plan for all those three aspects in your organization. You need a clear plan to remove existing cultural barriers while encouraging some new cultural behaviors among your people. Sure you are looking for more agility and effectiveness of day-to-day activities around the organization then, you should also consider redesigning and optimizing your current processes. Either you should have a practical method for migrating your legacy architecture and technologies you are using now to modern ones.

At this point we can quote one the most comprehensive definition of DevOps from Gareth Daine, DevOps Evangelist: "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." Thus we can expect DevOps something like what we see in fast movers environment as a mix of People, Process and Technologies working perfectly together.

Is DevOps just an Automation?

It is ubiquitous if you ever heard that DevOps is all about tools and automation. There is a famous quote from Grady Booch, which says "a fool with a tool is still a fool." Unfortunately, many teams and organizations considered the tools and automation as a silver bullet to every problem, but as discussed they should reconsider what they believe. Tools can help but cannot do magic. They should also think about processes and human factors of the organization too.

It is more relevant to consider tools and automation as a facilitator and accelerator to what we do. So, if we have ad-hoc processes to manage instead of predefined ones, there is nothing for us to accelerate.

DASA DevOps Principles

Many definitions of DevOps exist, and many of them adequately explain one or more aspects that are important to find flow in the delivery of IT services. Instead of trying to state a comprehensive definition on our own, DASA prefers highlighting six DevOps principles they deem essential when adopting or migrating to a DevOps way of working.

1. **Customer-Centric Action**: It is imperative nowadays to have short feedback loops with real customers and end-users, and that all activity in building IT products and services centers around these clients. To be able to meet these customers' requirements, DevOps organizations require the guts to act as lean startups that innovate continuously, pivot when an individual strategy is not (or no longer) working and constantly invests in products and services that will receive a maximum level of customer delight.

2. **Create with the End in Mind**: Organizations need to let go of waterfall and process-oriented models where each unit or individual works only for a particular role/function, without overseeing the complete picture. They need to act like product companies that explicitly focus on building

working products sold to real customers, and all employees need to share the engineering mindset that is required actually to envision and realize those products.

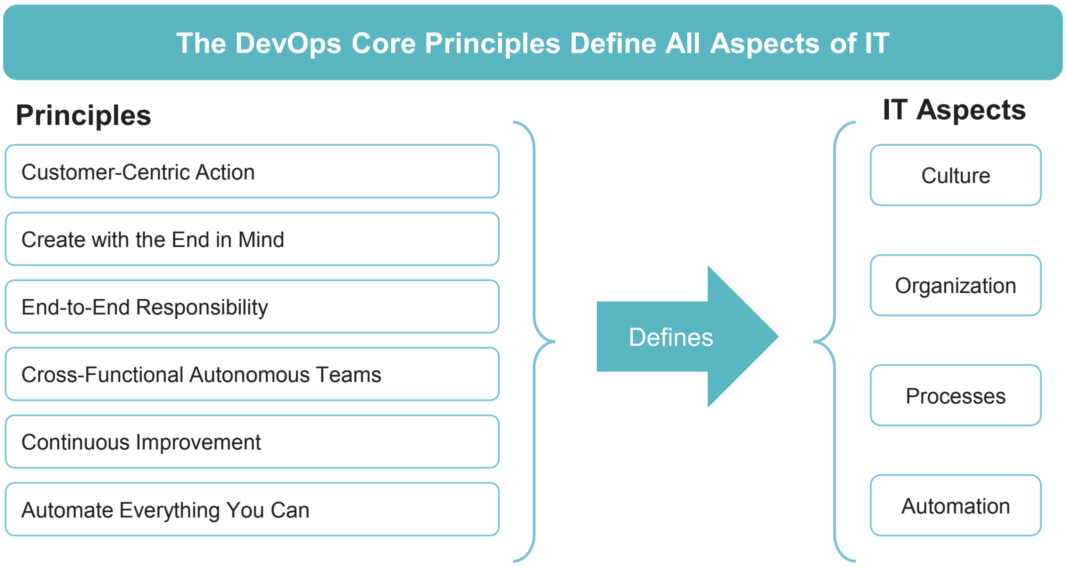
3. **End-to-End Responsibility**: Where traditional organizations develop IT solutions and then hand them over to Operations to deploy and maintain these solutions, in a DevOps environment teams are vertically organized such that they are fully accountable from concept to grave. IT products or services created and delivered by these teams remain under the responsibility of these stable groups. These teams also provide performance support, until they become end-of-life, which greatly enhances the level of responsibility felt and the quality of the products engineered.

4. **Cross-Functional Autonomous Teams**: In product organizations with vertical, fully responsible teams, these teams need to be entirely independent throughout the whole lifecycle. That requires a balanced set of skills and also highlights the need for team members with T-shaped all-round profiles instead of old-school IT specialists who are only knowledgeable or skilled in for example testing, requirements analysis or coding. These teams become a hotbed of personal development and growth.

5. **Continuous Improvement**: End-to-end responsibility also means that organizations need to adapt continuously in the light of changing circumstances (e.g. customer needs, changes in legislation, new technology becomes available). In a DevOps culture, a strong focus is put on continuous improvement to minimize waste, optimize for speed, costs, and ease of delivery, and to continuously improve the products/services offered. Experimentation is, therefore, an important activity to embed and develop a way of learning from failures is essential. A good rule to live by in that respect is “if it hurts, do it more often.”

6. **Automate Everything You Can**: To adopt a continuous improvement culture with high cycle rates and to create an IT organization that receives instant feedback from end-users or customers, many organizations have quite some waste to eliminate. Fortunately, in the past years, enormous gains in IT development and operations can be made in that respect. Think of automation of not only the software development process (continuous delivery, including continuous integration and continuous deployment) but also of the whole infrastructure landscape by building next-gen container-based cloud platforms that allow infrastructure to be versioned and treated as code as well. Automation is synonymous with the drive to renew how the team delivers its services.

DevOps and The IT Aspects


The DevOps Core Principles Define All Aspects of IT

Principles
Customer-Centric Action
Create with the End in Mind
End-to-End Responsibility
Cross-Functional Autonomous Teams
Continuous Improvement
Automate Everything You Can
Defines
IT Aspects
Culture
Organization
Processes
Automation

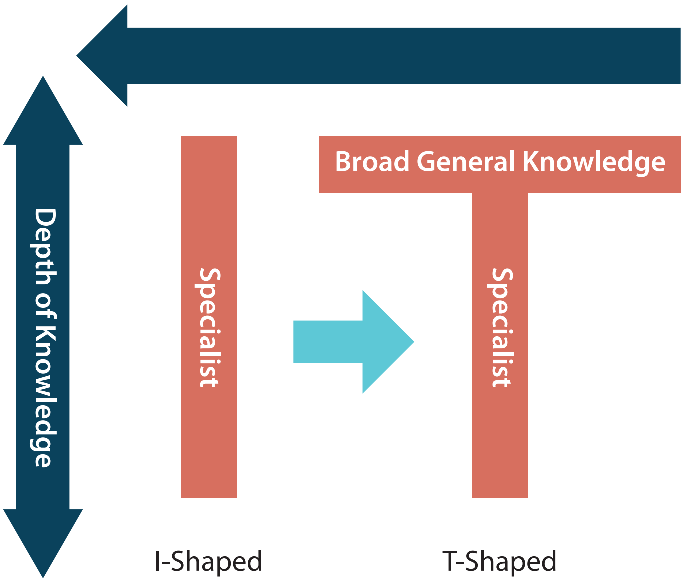
While organizations trying to use DevOps principles, they should consider that various IT aspects will be impacted and may need to be redefined. For example, if you want to support the "Cross-Functional Autonomous Teams" principle then you need to redefine your organizational structure, train team members with new things and concepts. Following "Automate Everything You Can" may require you to implement some new tools and also updating your current version of processes.

DevOps and People

As per discussed, DevOps needs organizational transformation activities, which will have impacts on individuals and teams simultaneously.

INDIVIDUALS

In DevOps, individuals cannot continue working in siloed roles with a limited narrowed down skillsets. DevOps is encouraging a T-shaped profile so individuals can understand what others do and support what they are doing. This will make team members work more closely and have a holistic view of what they will provide.



A T-shaped profile describes specific attributes of desirable workers in a DevOps environment. This kind of person will have expert knowledge and experience in a particular area, which will consider as a vertical bar of the T and also broad general knowledge, which helps him/her to collaborate with others and recognized as a horizontal bar of T.

DASA DevOps competence model categorized the skills and knowledge areas for each person in 12 subjects so that a T-shaped profile person will manage the required balance between these 12 subjects.

TEAMS AND ORGANIZATIONS

For the teams and organizations, there is much to think about. One of the most critical considerations about DevOps teams is their culture and people's behavior. If you look inside a DevOps organization you should see these aspects of cultural behaviors:

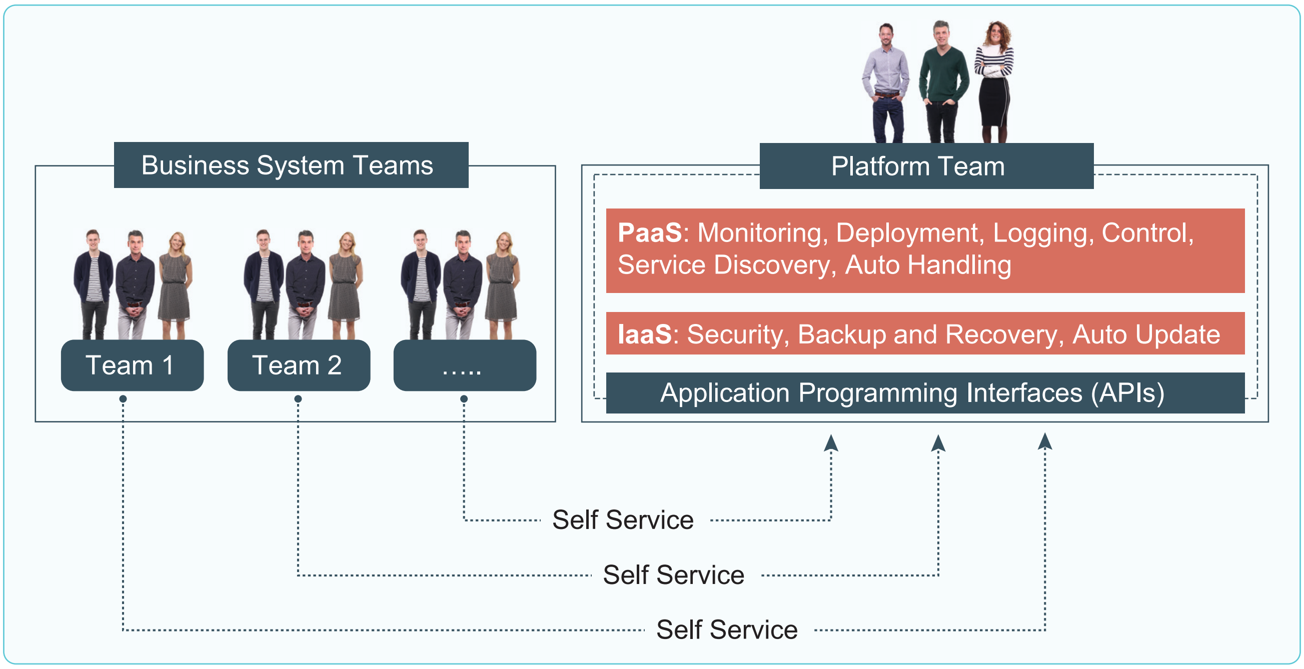
- Continuous Learning & Continuous Improvement
- Experimentation & Risk Taking

- Build Quality in
- An Engineering Culture
- A Culture of Effectiveness
- A Culture of Product Thinking
- A Culture of Taking Responsibility
- Inspirational and Fun Environment

Organizations cannot force these cultural aspects and tell people what to do, but they can act as a facilitator. They can grow any culture by providing the right context and taking some specific actions. For example, if they want their employees to be innovative and risk-taker, then they should give employees time for this and also set up a safe environment for the test.

Do you remember the term “wall of confusion”? We named an existing wall (physical or logical) between teams which blocks collaboration and information sharing “wall of confusion”, and this is how siloed organizations are working. While DevOps encourages organizations to break down these walls and avoid working in silos, organizations will need a new Organisational Model, which helps them to have fewer handover and many more collaborations.

DevOps teams need autonomy, and autonomy of teams is supported by the ability to operate as independent teams. The way to achieve autonomy through DevOps is setting up teams structured around distinct services and products. There are two types of teams, Business System Teams and Platform Teams. Business system teams focused on delivering business value through the delivery of software while Platform teams focused on the delivery and exploitation of platform services for DevOps Business System teams.


Business System Teams
Team 1
Team 2
…..
Platform Team
PaaS: Monitoring, Deployment, Logging, Control, Service Discovery, Auto Handling
IaaS: Security, Backup and Recovery, Auto Update
Application Programming Interfaces (APIs)
Self Service
Self Service
Self Service

In DevOps, a Business System team is responsible for the ‘health’ of a service. One of the crucial tasks when setting up such a team is to scope the responsibility of the team, which is vital for creating the required autonomy of the team. Without the insight on the customer, technical stack, and required knowledge, it is difficult to set up an autonomous Business System or Platform team.

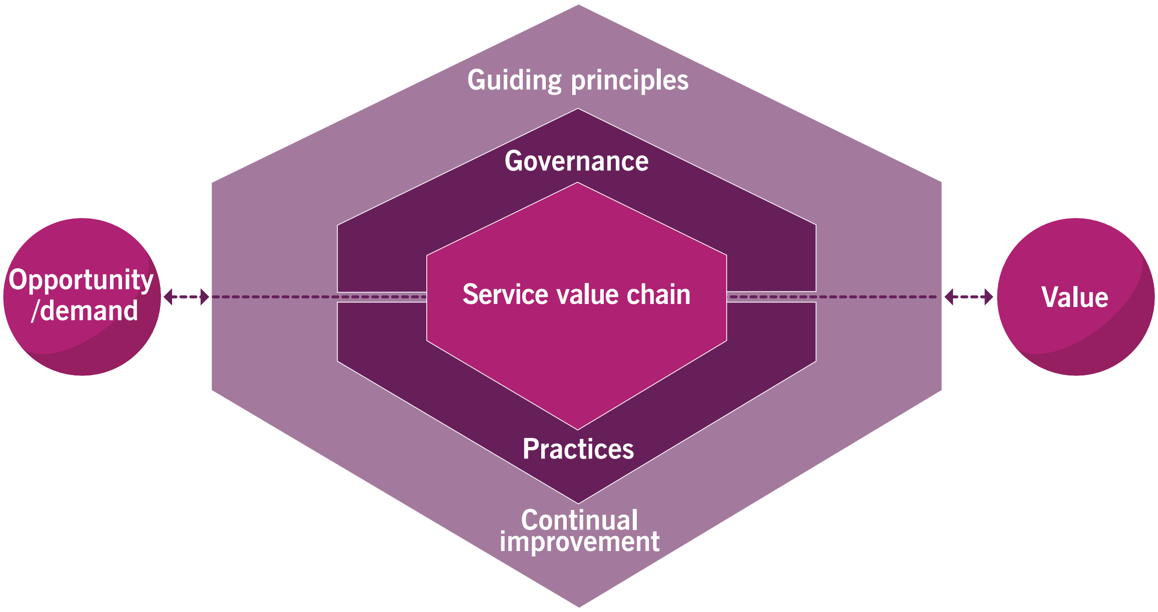
DevOps Practices

When you decided to adopt DevOps and its principles, you will need some detailed guidance through your way. This is when you can use some other rich frameworks and methodologies as your source.

ITSM

IT service management is an organizational capability for designing, planning, delivering, operating, and controlling information technology quality services offered to customers. ITIL is the most adopted framework and the source of practice for ITSM.

ITIL 4 as the latest release to this framework provides an end-to-end digital operating model for the delivery and operation of IT-enabled products and services and enables IT teams, to continue to play an essential role in the broader business strategy. ITIL 4 also provides a holistic end-to-end approach that integrates frameworks such as Lean, Agile, and DevOps.



Copyright © AXELOS Limited 2019. Used under permission of AXELOS Limited. All rights reserved.

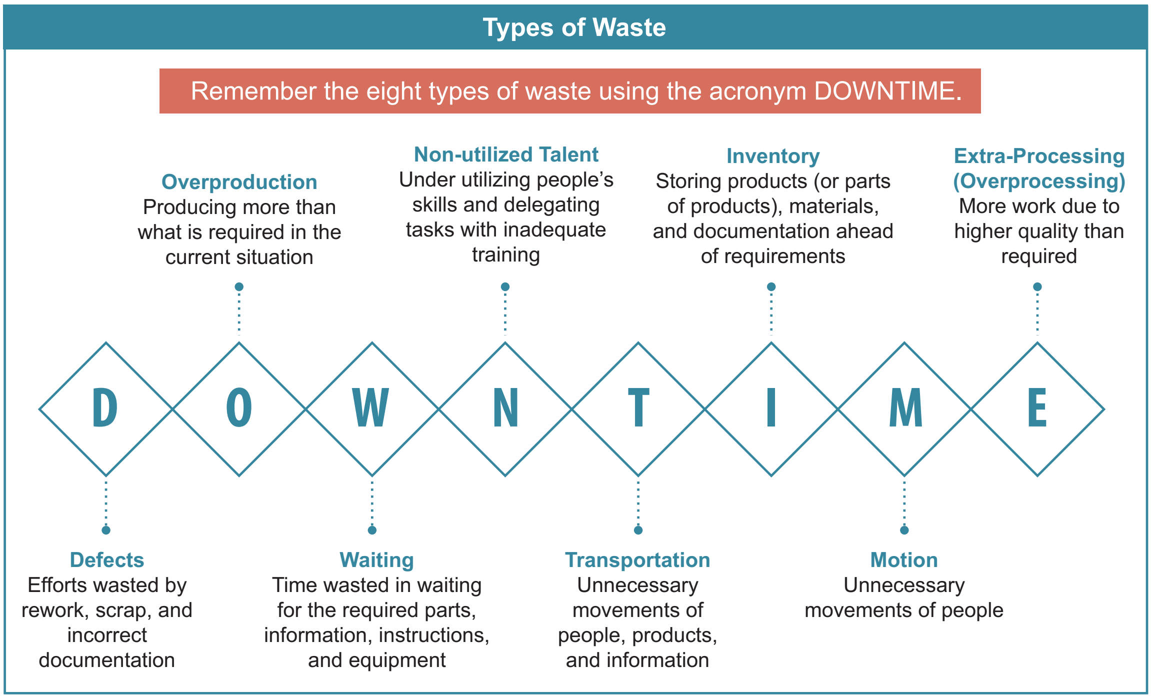
ITIL 4 includes 34 practices categorized in three different categories:

- **General management** practices are used in IT service management (ITSM) but are commonly used in general business areas as well such as Project Management, Risk Management, Organizational Change Management, etc.
- **Service management** practices have been developed in service management and ITSM industries such as Incident Management, Problem Management, Release Management, etc.
- **Technical management** practices have been adapted from technology management domains by expanding their scope from technology to ITSM which only have three practices named Deployment Management, Software Development And Management, Infrastructure And Platform Management

ITIL 4 will act as a toolbox at your hand and you can have everything you may need to manage a whole IT environment.

LEAN

Lean is a tested and proven method that uses a collection of tools to improve the way products and services are produced. It is also considered a mindset that pushes individuals to think about making the services better on a daily basis.



Types of Waste

Remember the eight types of waste using the acronym DOWNTIME.

Overproduction
Producing more than what is required in the current situation

Non-utilized Talent
Under utilizing people's skills and delegating tasks with inadequate training

Inventory
Storing products (or parts of products), materials, and documentation ahead of requirements

Extra-Processing (Overprocessing)
More work due to higher quality than required

Defects
Efforts wasted by rework, scrap, and incorrect documentation

Waiting
Time wasted in waiting for the required parts, information, instructions, and equipment

Transportation
Unnecessary movements of people, products, and information

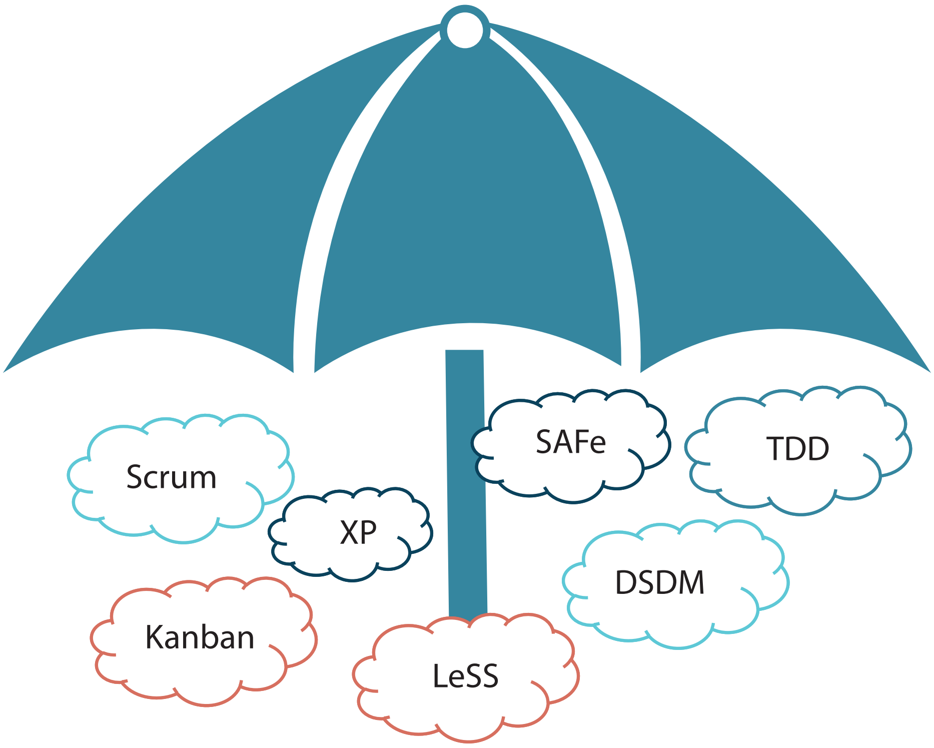
Motion
Unnecessary movements of people

Taiichi Ohno, considered the father of Lean, was instrumental in developing the "Seven Wastes" model which has become the core in many academic approaches. An eighth waste, non-utilized skills, is commonly overlooked, but it is also a waste. The eight wastes that can be identified from a Lean perspective are:

- Defects
- Overproductioncessing
- Waiting
- Non-utilized skills
- Transportation
- Inventory
- Motion
- Extra-Processing (or Waiting
- Overprocessing
- OverproOverprocessing)duction

AGILE

Agile is a time-boxed and iterative approach of software delivery. It aims to build software incrementally from the start of the project. Agile relies on adaptive planning and iterative development and delivery. It focuses primarily on the value of people in getting the job done effectively. Agile is not a method in itself. Agile is the umbrella, based on the values & principles from the Agile Manifesto, which comprises several methods and frameworks.



- By far, **Scrum** is the most used and documented Agile framework worldwide
- **Kanban** is gaining more and more ground in Agile teams as a powerful method for IT delivery and support.
- **eXtreme Programming** (XP) provides Agile teams with key technical practices (such as CI, pair programming)
- **DSDM** (Dynamic Systems Development Method) was one of the first agile methods for software development (originated around 1994)
- **SAFe** (Scaled Agile Framework) and LeSS (Large Scale Scrum) are both influential frameworks enabling agile at scale in large enterprises.
- **TDD** (Test Driven Development) focuses on the practice of first creating tests before writing any code. Once the test cases have been written, the software (code) is improved until the tests pass, after which the code is refactored. This is a fundamental method enabling true Shift Left in testing.

IT ARCHITECTURE

Architecture in IT is like what it means in building an apartment. Architect plans about the foundation of what is going to build and its structure based on requirements like how many floors will this apartment has.

Architecture is one of the essential aspects you should consider in DevOps. Architecture is basically related to the technology dimension of DevOps. When you are working on a product, you should think like solving a puzzle and architecture will help you decide to put each puzzle part in the right place based on your needs.

For years, organizations experienced different patterns of architecture such as Monolithic Application, Multilayered Architecture, Component-based Architecture, Service-oriented Architecture, and Microservices Architecture. Among all these, MSA (microservices architecture) fits the best in DevOps environments.

CONTINUOUS DELIVERY AND AUTOMATION

As per discussed DevOps encourages automation a lot as we have a principle for it "Automate everything you can". Continuous Delivery, Integration, and Deployment are three different aspects of each software life-cycle management which DevOps teams invested on a lot to automate. Let's review some definitions of these three:

> **"Continuous Delivery is about putting the release schedule in the hands of the business, not in the hands of IT**. Implementing Continuous Delivery means making sure your software is always production-ready **throughout its entire lifecycle – that any build could potentially** be released to users at the touch of a button using a fully automated process in a matter of seconds or minutes."
>
> Jez Humble

> "**Continuous Integration** usually refers to integrating, building, and testing code within the development environment. "
>
> Martin Fowler

> "**Continuous Deployment** is subtly different from Continuous Delivery in that release are automatically pushed into production when all tests pass. In Continuous Delivery, release is a human decision."
>
> Dave Farley

Teams that have adopted these three will be highly effective and respond to changes requirements more quickly than others do these things manually or by human intervention.

CLOUD COMPUTING

Cloud computing is a new way of IT service delivery such as storage, databases, software, processing, and others. Cloud service providers will charge their customers for the services they provided, and most of the time charges will be based on customer usage of the service. Cloud services which support by new technologies and architecture can be categorized in these three categories:

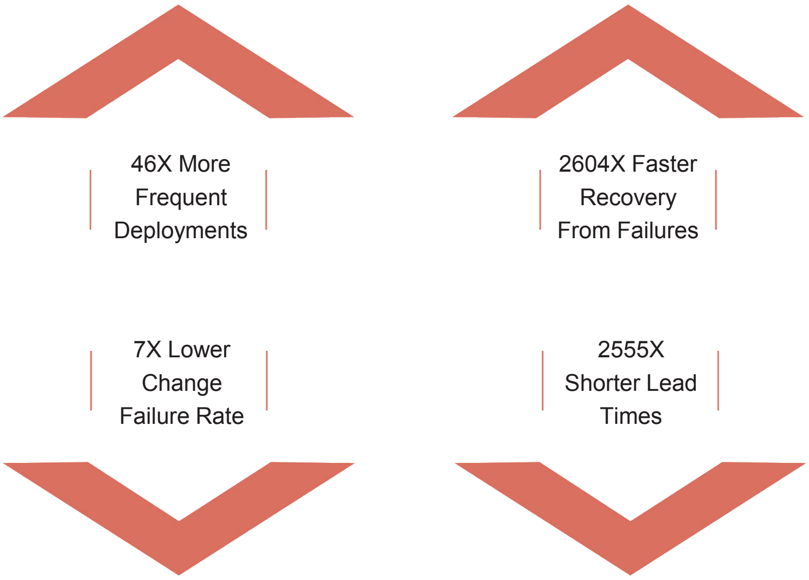
- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

Cloud computing changes the way businesses think about Costs, Flexibility, Scalability, Reliability, and other important subjects. Cloud services can be deployed in four ways:

- **Public clouds** provided by third-party cloud service providers such as Amazon, IBM, Google, Microsoft, etc. delivering servers, storage, processing capabilities over the Internet.
- **Private cloud** usually deployed for internal usage of a corporation for hardware and infrastructure resources optimization.
- **Hybrid clouds** are a combination of public and private clouds. This deployment of cloud uses an advanced technology that allows data and resource sharing between those two types of clouds.
- **Community clouds** are another way of cloud service deployments, which allows several companies (basically in the same group or any other kind of relations) to use shared resources.

Insights of DevOps Adoption

It's been a while that Puppet labs release a report each year named *State of DevOps Report*. In this report, we can see how high performing IT organizations that used DevOps principles are doing much better than their competitors. Report of the year 2014 indicated that those high performing IT organizations deploy 30 times more frequently, have 200 times shorter lead times, have 60 times fewer failures, and also recover 16 times faster! These numbers are fascinating.


46X More Frequent Deployments
2604X Faster Recovery From Failures
7X Lower Change Failure Rate
2555X Shorter Lead Times

Source: State of Devops report 2017

These kinds of researches showing that companies adopting DevOps are benefiting in various areas like:

- Improved speed to market
- Continuous Integration and delivery
- Higher quality, fewer failures
- Fewer failures
- Higher stability

- Innovation and creativity
- Increased employee engagement and job satisfaction
- Breaking down silos and eliminating waste
- Resource and cost reduction

DevOps Journey of Easy Journey Airways

Dream journeys do come true!

The course includes a fictitious case study, “Easy Journey Airways.” The case study aims to enable you to apply the DevOps concepts in a real-life situation. The various assignments included in the course are based on this case study and will help you experience the DevOps transformation of EJ Airways.

To successfully perform the assignments during the course, you should go through the case study in advance. The Course Book contains the case study as Appendix A.