

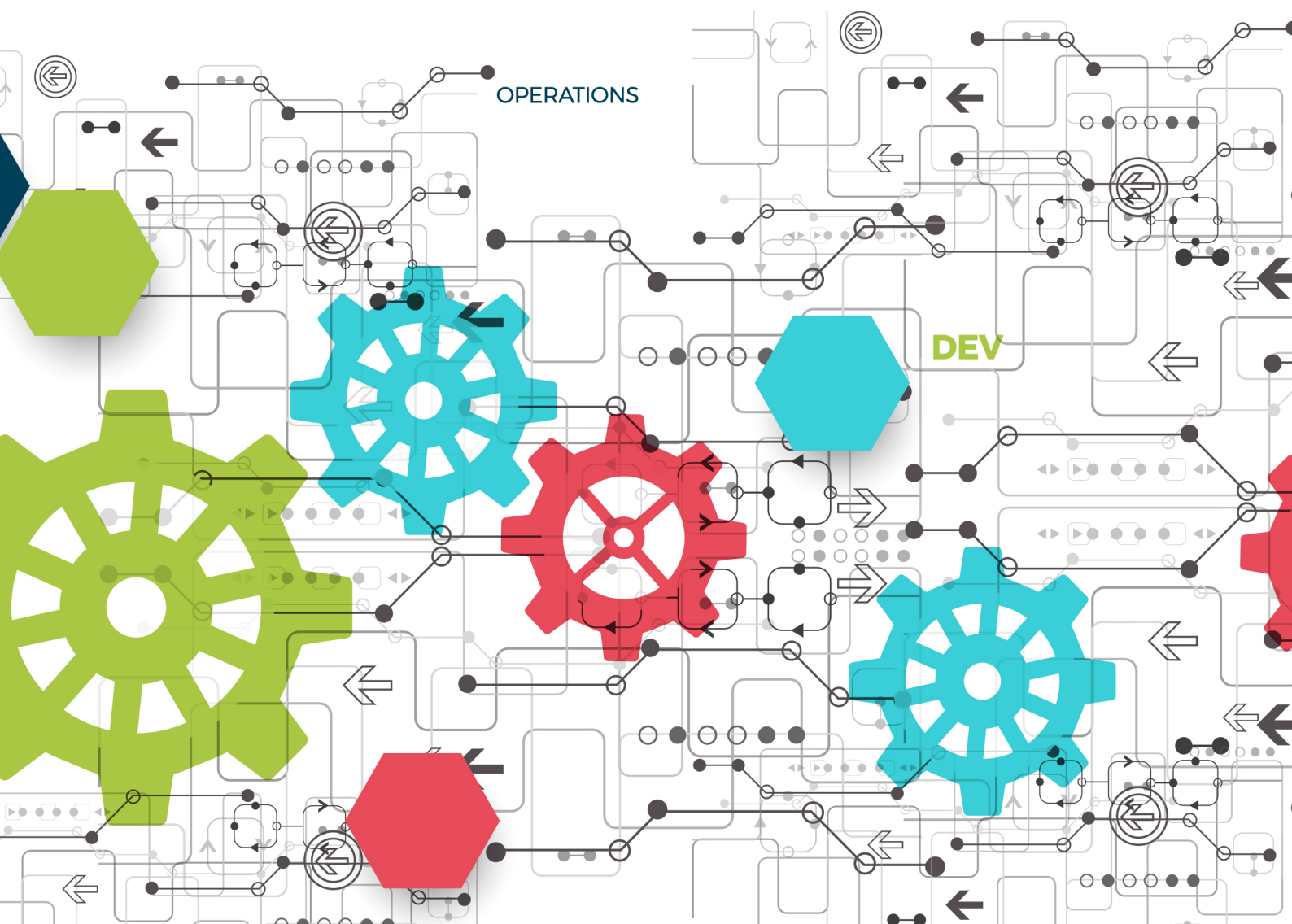


**B.Tech** Computer Science  
and Engineering in DevOps

# DEVOPS OVERVIEW

MODULE 1

## Definition of DevOps



# Contents

<b>Module Learning Objectives</b>	<b>1</b>
<b>Module Topics</b>	<b>2</b>
<b>1. Challenges of Traditional IT systems</b>	<b>3</b>
<b>2. Emergence of DevOps</b>	<b>8</b>
<b>3. Definition for DevOps</b>	<b>10</b>
<b>4. CAMS Model – Gene Kim</b>	<b>16</b>
<b>5. DevOps and Agile</b>	<b>19</b>
<b>In a nutshell, we learnt:</b>	<b>26</b>

## MODULE 3

# Definition of DevOps

### Facilitator Notes:

Introduce the module to the participants and tell that you will talk about the introduction to DevOps in this module.

You will learn about the definition and overview of DevOps in this module.

### Module Learning Objectives

At the end of the Module you would be able to learn the following

- Enumerate the pitfall of traditional IT systems and its processes
- Describe the evolution of DevOps
- Explain the core concepts of DevOps
- Enumerate the core principles of DevOps
- Understand the benefits of embracing DevOps
- Identify the need for building a business case for DevOps



**Facilitator Notes:**

Summarize the module objectives for the participants.

Reiterate that at the end of the module you will be able to:

- Enumerate the pitfall of traditional IT systems and its processes
- Describe the evolution of DevOps
- Explain the core concepts of DevOps
- Enumerate the core principles of DevOps
- Understand the benefits of embracing DevOps
- Identify the need for building a business case for DevOps

**Module Topics**

The following topics that will be covered in the module:

1. Challenges of traditional IT systems & processes
2. History and emergence of DevOps
3. DevOps definition and principles governing DevOps
4. DevOps and Agile
5. The need for building a business use case for DevOps

**Facilitator Notes:**

Tell the participants that they will be discussing the following topics:

- History & emergence of DevOps.
- Early adopters of DevOps.
- The definition of DevOps.
- The similarity between Agile, Lean & DevOps.
- Fundamental principles governing DevOps.
- Explain each principle in detail.

The following topics will be discussed:

History & emergence of DevOps.

- Early adopters of DevOps.
- The definition of DevOps.
- The similarity between Agile, Lean & DevOps.
- Fundamental principles governing DevOps.
- Understanding each DevOps principal in detail.

## 1. Challenges of Traditional IT systems

Traditional IT systems follow outdated processes that are not suited for products of today. This leads to many challenges for the organization

<b>Poor Quality of products</b>	<ul style="list-style-type: none"> <li>→ Organizations create and deliver products that offer less or no value to the end customer leading to product – market gap</li> <li>→ Products fail to meet the expectations of the customer, who seek high performance, ease of use sturdy</li> </ul>
<b>Irregular release or updates</b>	<ul style="list-style-type: none"> <li>→ Software products need continuous updates and new releases to meet customer's growing demands.</li> <li>→ Organization following traditional model are unable to update at shorter time spans, thereby becoming obsolete to the customer</li> </ul>
<b>Product backlog</b>	<ul style="list-style-type: none"> <li>→ Most of the traditional IT organizations follow archaic systems that results in huge product backlog and delay in delivery of products</li> <li>→ Product backlog further leads to business loss</li> </ul>
<b>Outdated processes</b>	<ul style="list-style-type: none"> <li>→ Traditional systems follow outdated processes leading to loss of effort, time and money for the organization</li> <li>→ Organization adhering to traditional practices are siloed and closed with little interaction between different functions leading to poorly built products</li> </ul>

### Facilitator Notes:

Give a brief background to the participants on the evolution of the software industry. From mainframes computers to personal computer to mobile devices, walk them through the changes that have slowly pervaded their lives and changed the way we operate and function forever.

Talk about the evolution of software product building, which has moved from being an enterprise-ready application to customer-driven applications. The same processes that were employed during the monolithic days are not suited for today's applications.

Explain the internal challenges such as lack of collaboration between teams, budget constraints, decision-making processes that deeply embroil organizations still following traditional systems. Also, talk about the external challenges such as staying competitive, offering products that meet the growing customer needs.

Talk to participants about the software delivery lifecycle and the different stages that are involved in building and delivery of a product. Draw a comparison of how products such as Lotus Notes, an enterprise application remained largely unchanged as opposed to customer applications such as an Uber or WhatsApp that are used by millions of people.

Talk to participants about how traditional organizations were structured in silos completely cut away from each other. Product development is a complex activity and demands collaboration of different functions and teams to create world-class products. Discuss the apps that are used by participants on a daily basis such as WhatsApp or Uber which cater to millions of people across geographies. A siloed organization with limited communication does not offer an environment that can build these world class apps.

Introduce participants to the traditional processes in how businesses operated. A linear method of product building where each department was responsible for different functionalities with limited interaction. Explain the effects of this such as poor feedback loops, lack of knowledge of product across teams, lack of uniformity in process etc.

In the last two decades, technology has undergone sea change starting from the advent of large-sized computers to microprocessors to personal computing to more recently the internet and now mobile phone. The rapid change has redefined the age-old wisdom that companies have been following and now forces them to change. Software products were in the past built for large enterprises that were used within the organization to manage their processes and products. Following the advent of the internet, this has dramatically changed and has moved from being enterprise-centric to a user-centric model, what businesses define as a B2C model. IT & Software is all-pervading and is prevalent across businesses, different industry segments and even our homes. For instance, think of the number of devices, an individual uses on a daily basis.

Traditionally organizations were highly linear, where information dissipation was very slow and decision making was limited to the management. Today this is not the case. Customer feedback is actively followed and implemented on a constant basis in order to stay relevant and competitive.

Organizations that follow the traditional way of software development work with strict principles, and in these organizations, the Development and Operations teams function as two separate entities. Development team tends to be driven by how many new functionalities can be churned out in a given time, therefore change is its incentive. Operations team on the other hand, tends to be driven by stability of the status quo and its incentive is therefore resisting change.

The lack of flexibility in the traditional organizations to switch from a traditional mindset has led to the creation of products that are irrelevant. Organizations constantly battle to create the right products that fit the customer or market needs within a given time frame and allocated cost & budgets. In today's day and age, the barriers between the customer and organization are diminishing rapidly and customers constantly look for products that best suit their needs. Customer's loyalty is directly determined by the organization's ability to offer products that satisfy their need and offer value for their money. Traditional systems have a huge disconnect and follow age-old processes and invest more time in processes than the product itself.

### ***Irregular release or updates***

Traditionally enterprise applications such as IBM Lotus Notes would have a software update once in few months. Today such a long delay cannot be tolerated as customer's needs and demands have vastly changed. For instance, how many updates does your phone undergo on a given day? How many mobile applications update on a daily basis? New-age companies push product releases and updates few times on a given day.

This is a critical factor to stay competitive and relevant to their customer needs. Product building is a time-consuming activity. However, in today's day and age, staying relevant and steadfast is most critical for the survival of an organization. Many of the products lose their market and become obsolete within a few months of its launch. They are replaced with new products and new players at a merciless pace.

### ***Product Backlog:***

Traditional organizations follow traditional processes and are structured very centrally. Information dissipation is very slow leading to bottlenecks across the organization. Decision making is very slow and this impacts the product building and delivery on a large scale. The pace and processes of traditional IT systems can lead to a huge backlog in terms of product updates and product releases directly affecting the organizational growth. This can lead to losing to marquee customers and huge monetary loss for the company.

For example, a ride-hailing application such as Uber is used by millions of users, both riders and drivers across geographies. This means that companies operate out of different geographies across different cultures catering to many different languages and different time zones. In such a scenario, how can a siloed organization with barriers create a product to suit the customer's need? The business use case pushes organizations to rethink the way they operate and disrupt the traditional models.

### ***Outdated processes:***

Organizations continue to follow traditional IT processes that are obsolete in today's day and age. Considering the type of products companies build today, they need to innovate and deliver products at a faster-pace and time compared to earlier generations. This pace can only be achieved by getting rid of traditional processes and adopting new processes that are best suited for a fast-paced, competitive environment. For example, a software engineer in Bangalore operates in a decentralized environment interacting with team members across different geographies collaborating to put together a product that is used globally.

## **1.1 Disconnect between Development & Operations teams**

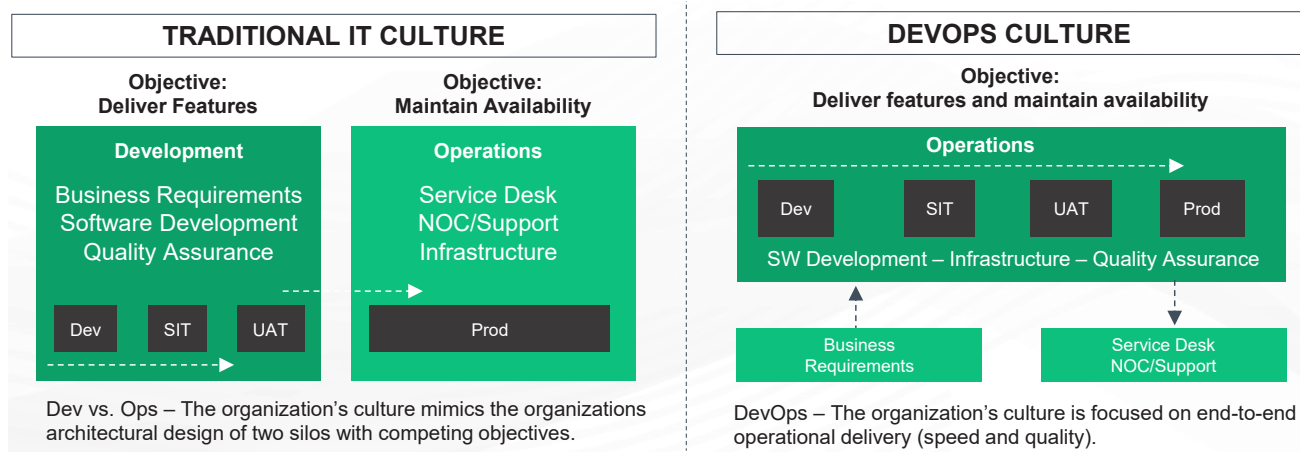
---

The primary challenges between two teams are

- Lack of collaboration
- Disintegrated processes
- Difference in tools & implementation processes



- Disinterest in learning new tools
- Difference of opinion
- Work loss
- Poor feedback system



### Facilitator Notes:

Application building has undergone tremendous changes in the last decade. Computer applications are defined by logic. Applications, file systems, databases and later cloud all adhere to a logic. Traditionally, how applications interact with devices, how data centres were built to host and manage these applications remained largely the same. However, this is not the same as the applications that are built today. Applications today are built to run across devices, with different operating systems, hosted in cloud or data centres. The juxtaposing means that the lines between functions are blurring by the day. Development is no more purely development and operations is no more purely operations. The 'Developers' need to know operational aspects to build applications and the 'Administrators' need to know development to guide, offer feedback, run and manage the application on a daily basis.

Traditional methods and processes did not demand the coming together of different functionalities.

In organizations, development & operations, the teams work independently which lead to a lot of friction internally which in turn impacts the business growth. The lack of knowledge sharing, poor handing over processes and miscommunication leads to innumerable delays and business loss for the organization. Application building has undergone tremendous changes in the last decade. Computer applications are defined by logic. Applications, file systems, databases and later cloud all adhere to a logic. Traditionally, how applications interact with devices, how data centers were built to host and manage these applications remained largely the same. However, this is not the same of applications that are built today. The Applications, today are built to run across devices, with different operating systems, hosted in cloud or data centres. The juxtaposing means that the lines between functions



are blurring by the day. The 'Development' is no more purely development and the 'Operations' is no more purely operations. The Developers need to know operational aspects to build applications and administrators need to know development to guide, offer feedback, run and manage the application on a daily basis.

The traditional methods and processes did not demand the coming together of different functionalities.

In organizations, development & operations teams work independently and this leads to a lot of friction internally which in turn impacts the business growth. Lack of knowledge sharing, poor handing over processes and miscommunication leads to innumerable delays and business loss for the organization.

**Lack of collaboration:** Both development and operations function as separate teams and do not have the attitude or interest to collaborate with each other

**Disintegrated processes:** The processes followed by both the teams are naturally different considering the nature of the work. This leads to confusion and the processes of both teams do not integrate well

**The differences in tools & implementation processes:** Both Development and Ops teams using different tools and processes can lead to constant errors and bug fixes in the production environment. Similarly, both teams using different implementation processes to execute the same work can cause incompatibility also.

**Disinterest in learning new tools:** Both teams have their own tools and processes and consider their methods superior to the other. This leads to lack of cohesion and waste of time.

**The difference of opinion:** The development teams in every organization constantly look out for new technology or updates and continuously make changes, whereas operations teams believe this leads to instability.

**Work loss:** The constant back and forth between the teams lead to loss of work and in turn business loss.

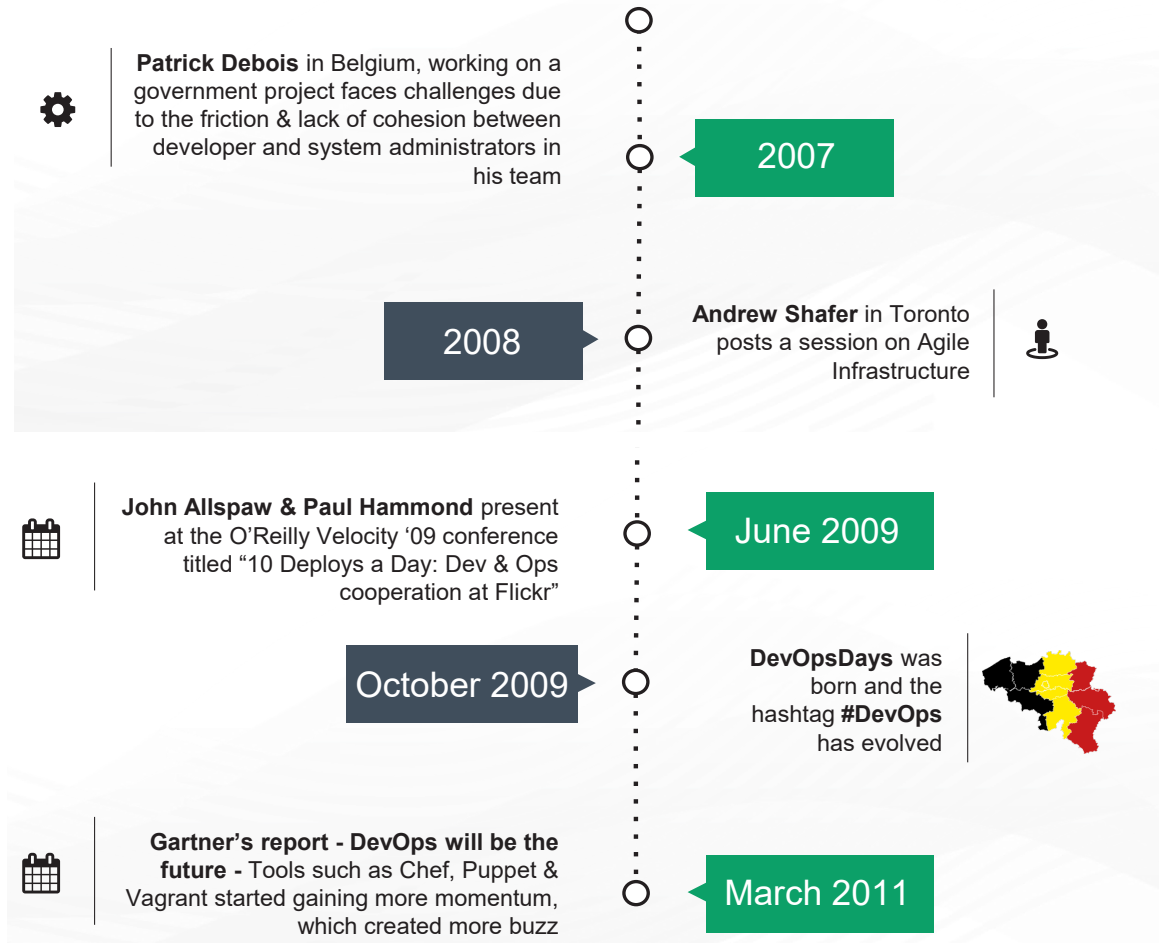
**Poor feedback system:** The Ops and Development teams fail to collaborate and lack any structured process. This lack of processes also leads to poor feedback systems which in turn, lead to organizational gaps.

Because of the confusion and the disconnect that happens between Dev and the Ops teams, industry experts desperately wanted to find a common ground, that solves all these issues. This frustration led to the birth of DevOps movement, and the life of IT development and operations teams became a lot easier. Organizations found the concept to be useful and slowly started adopting it as a culture. The wall between the dev and ops teams started tearing down, which resulted in better quality products delivered in shorter time periods.

## 2. Emergence of DevOps

Development and operations teams have functioned as two different entities without any collaboration. This mindset changes due to the challenges that teams faced across organizations

### EMERGENCE OF DEVOPS



#### Facilitator Notes:

Explain in detail how DevOps came into being and how it reached the popularity it has gained today. With the advent of the internet, applications have undergone tremendous changes in terms of performance, usability and the benefits they offer compared to traditional IT software. This has further to relooking at IT infrastructure where large data centres have today been replaced by cloud. Similarly, users, today expect a response time of 1 second or even lesser as they are highly dependent upon them. This led to software- defined infrastructure which in turned paved way for DevOps.

DevOps is to the software industry what industrial revolution is to manufacturing.

Professionals across organizations were frustrated at the lack of cooperation between development and operations teams which led to the loss of time and money. Patrick Debois in Belgium was one such consultant who was working on a major data centre migration project for their government faced numerous frustrations due to the sheer lack of cooperation from Development and Operations teams and the time it took. He was looking for a more efficient way of running the project and how both dev & ops teams could work in cohesion.

**2008:** In August 2008, Andrew Shafer of Toronto publishes a talk on “Agile Infrastructure” at a local event. This, however, does not happen due to the poor turnout. Patrick from Belgium on learning about the discussion reaches Andrew and both of them vent their frustrations over continuously managing dev & ops teams and its repercussions. The meeting gives rise to Agile Administration Group on Google Groups.

**2009:** John Appspaw & Paul Hammond delivered a talk titled “10 Deploys a Day: Dev & Ops cooperation at Flickr” at the Velocity Conference in San Jose.

**October 2009:** Discussions led to more discussions and DevOpsDays was first born in Ghent, Belgium, it was a two-day conference held on 30 and 31, October 2009. The event was very well received and was attended by the professionals across Development and Operations team and was a grand success. This led to more and more ‘DevOpsDays’ across locations and across other countries creating a wave of DevOps.

**DevOps tools:** The momentum continued to rise and the professionals across the globe started contributing. Organizations started turning their focus towards DevOps tools such as Chef, Puppet & Vagrant, and more and more organizations started adopting these tools. This created more buzz in the industry. In March 2011, Gartner published a report, to establish the significance of DevOps and how it will become the future mandate for organizations to perform better and smoother and deliver better quality IT products. Organizations, vendors and analysts started paying more attention to the movement and DevOps as a culture was coming into existence.

## 2.1 Early Adopters of DevOps

---



**Facilitator Notes:**

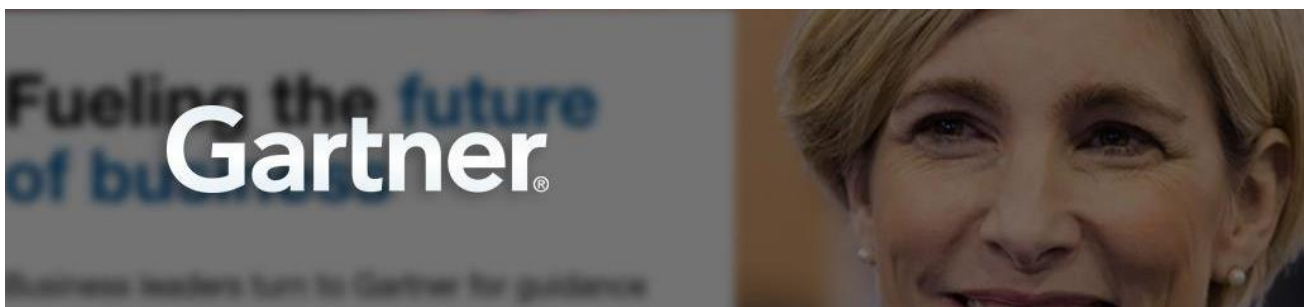
Tell the participants that some of the early adopters of DevOps are:

- Google
- Amazon
- Facebook
- Uber
- Netflix

Some of the early adopters of DevOps are:

- Google
- Amazon
- Facebook
- Uber
- Netflix

### 3. Definition for DevOps



*“DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology — especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective”*

Source adopted from <https://www.gartner.com/it-glossary/devops>

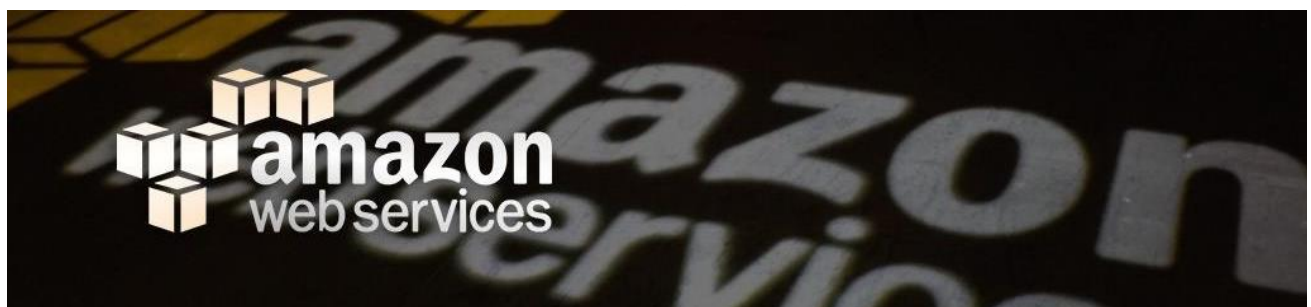
**Facilitator Notes:**

Explain each of the above mentioned definitions, to participant, in details.

DevOps has been widely accepted and embraced by the various organizations worldwide. Each organization defines DevOps differently, based on their experience and the reception that people gave.

Gartner defines DevOps as 'a change in the mindset of IT thought leaders, breaking age old traditional practices and adapting to Agile & Lean principles that emphasise on customer value.'

DevOps focuses on the tangible and intangible aspects of an organization. The People, culture, collaboration and communication among different functions within the organization come together to deliver efficient products. In the process, DevOps uses tools and technology, to automate processes that are manual and inefficient.



*DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.*

Source adopted from <https://aws.amazon.com/devops/what-is-devops/>

Amazon Web Services, the leading public cloud- provider defines DevOps, as a combination of processes, practices, techniques and tools that the organizations use to facilitate speedy delivery of the products. These practices also help the companies to innovate, ideate and iterate at a faster pace and release new features and launch products at a shorter time- intervals as opposed to the traditional systems. DevOps is an important way for the organizations to stay relevant and competitive.



*DevOps (a clipped compound of "development" and "operations") is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives*

Source adopted from <https://en.wikipedia.org/wiki/DevOps>

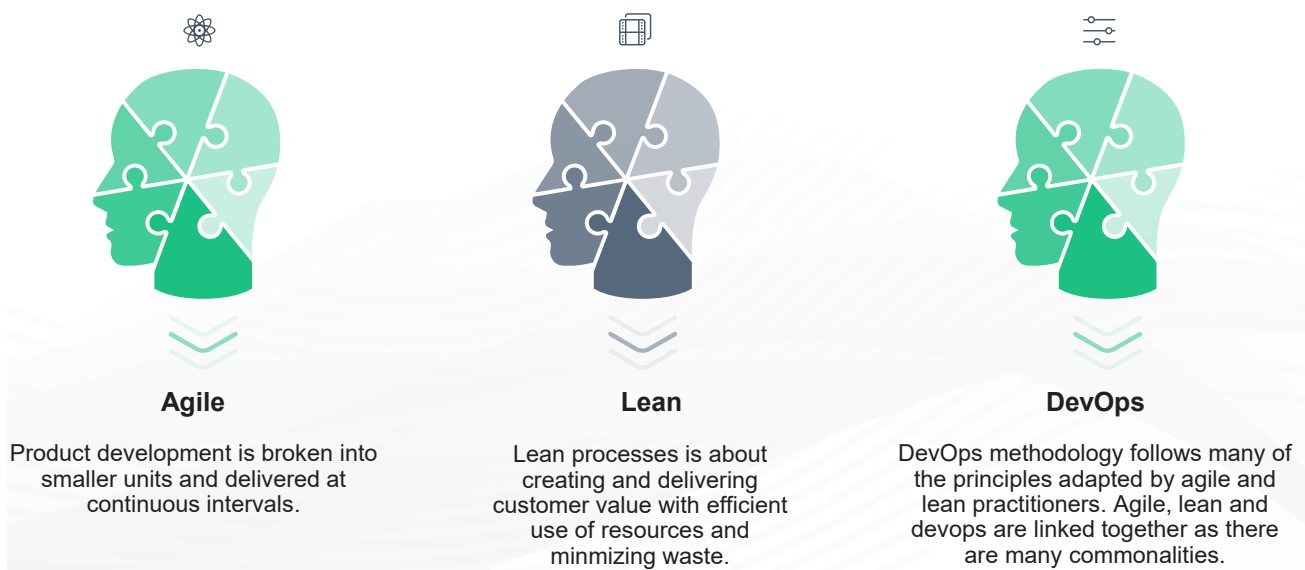
**Facilitator Notes:**

Explain each of the definitions mentioned above in detail to participants.

Wikipedia's definition of DevOps is a combination of Gartner's & Amazon's definition of 'DevOps'.

Wikipedia defines DevOps, as a coming together of the development and operations.

### 3.1 Agile, Lean & DevOps

**Facilitator Notes:**

The Agile, Lean & DevOps processes have one primary goal: To create the customer value.

Explain the three processes, to draw similarities, for the participants to better understand the principles DevOps is based on.

**Agile:** The Agile methodology is a break-away from the traditional methods of product-building to enable faster delivery cycles. The Agile methodology breaks product delivery into smaller iterations, offering a complete product, every time there is a product release. Agile also brings together different teams within an organization for faster delivery of the product.

**Lean:** The primary motive of lean processes is to deliver customer value with optimum utilization of process and to eliminate waste. Lean processes involve an overhaul of all processes in the product life-cycle eliminating any waste. Lean Startup tends to focus more on a method for product development.

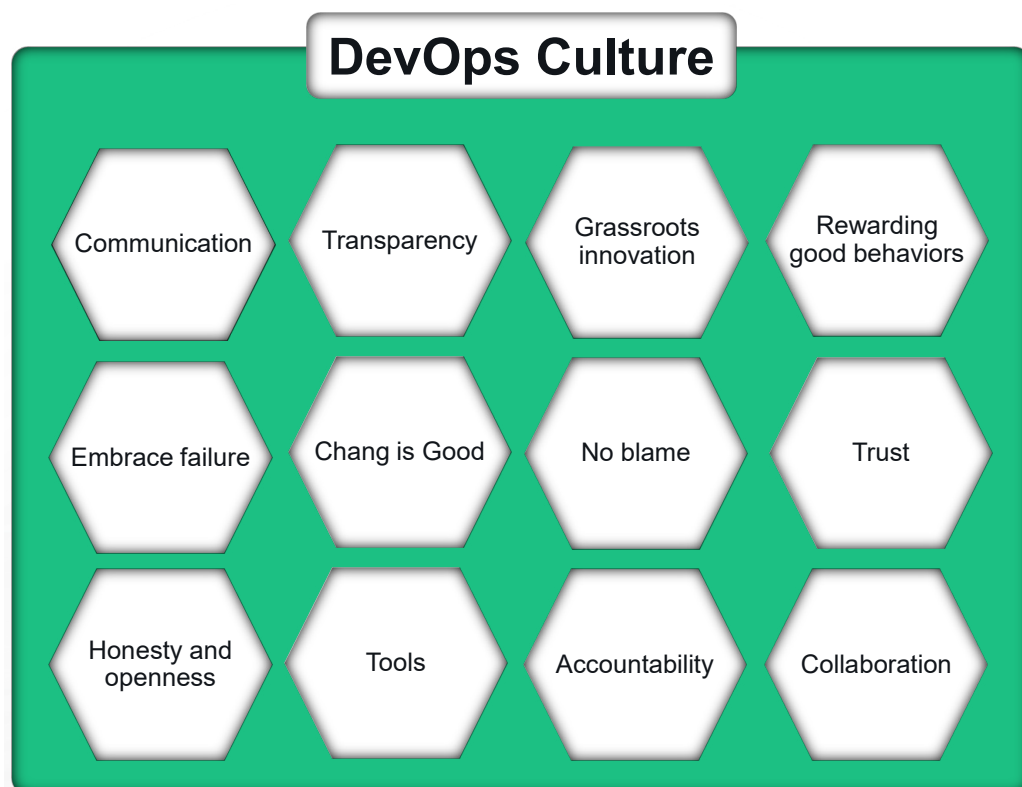
**DevOps** imbibes these philosophies and embraces a culture that propagates collaboration & seamless communication between the teams to speed up delivery of products. DevOps is one of the central pillars on which many of the new breed of IT organizations realize a new modus operandi for delivering IT services. Adopting DevOps across the entire organization helps organizations redesign



their business and IT departments. DevOps thus becomes a new operating model that bids adieu to traditional demand-supply models, centralized IT operations, and complex value streams with an excess of hand overs, waste and error-prone manual activities that do not deserve the label 'engineering'. DevOps is the most holistic way and more likely to take cultural aspects and the existing operation into consideration.

### 3.1.1 DevOps – A Culture & Its Benefits

- DevOps is not a framework or a standard to be implemented but a culture that pervades the organization and enables speedy delivery of products and services.
- DevOps is an amalgamation of development and operations to create a more efficient, foolproof process that supports collaboration and teamwork leading to high performance.



#### Facilitator Notes:

Explain the core foundation of DevOps and the necessity for the implementation of DevOps. Then address the different issues that organize the face in the battle, to stay competitive and relevant.

DevOps is a synopsis of Lean & Agile practices coming together, to offer faster delivery of the products. It is not a standard or an IT framework that the organizations implement but more of a philosophy and culture that forms the core of how the organizations' function.

Traditional organizations followed outdated processes that offered no room for collaboration. They had rigid structures and the decisions were dictated top-to-down making, day to day operations



of a business, cumbersome and inefficient. DevOps disrupts the traditional models by following a decentralized organization, where the environment encourages and supports open communication, teamwork and an open culture. This leads to many benefits to the organization.

### 3.1.2 DevOps – Benefits

DevOps adoption offers numerous benefits to organizations at large

Reduction of lead time	DevOps propagates efficiency and reduces the lead time for product delivery
Reduction in failure of product & its releases	A stable environment facilitates stable product building. With the amalgamation of development and operations, processes are streamlined and product releases happen at shorter timelines
Collaboration	DevOps is about collaboration. Different teams such as development, operations, quality assurance, testing and support coming together and work cohesively and deliver products in a shorter time span than traditional systems
Scalability	Automation helps you manage large infrastructure efficiently and devops adoption helps companies scale faster
Reliability	As a culture, devops strives for stability and fault proof products. DevOps creates efficient processes that eliminate errors and time loss
Optimum utilization of cost and resources	Implementing a devops culture means a thorough study of end to end lifecycle and removing bottlenecks if any. The primary goal is to offer customer value and continuously eliminate wastage

#### Facilitator Notes:

Tell the participants that you will talk about the benefits of DevOps.

DevOps has the following benefits:









- **Reduction of lead time:** DevOps brings together the development, operations and QA teams together and creates a process that allows collaboration. This reduces the feedback loops between the teams leading to shorter delivery cycles.
- **Reduction in a failure of the product & its releases:** Since the different functions involved in product-building & delivery of the work are in unison, under efficient processes, the room for error is reduced drastically in comparison to the traditional systems.
- **Collaboration:** In a pre-DevOps scenario, the development teams build and push the product to QA team for testing. They, in turn, test the product and do the test runs to identify issues or what we call the bugs in the Developer-Jargon. This is then brought back to the development team to rework upon and this loop continues. Once this comes to closure, the product is handed over to the operations. The next round of challenges starts for the operations' team in maintaining and running the product on a daily basis. The lack of a standard product manual, poor handing-over process and constant fire-fighting leads to excessive delay. This can only be overcome by implementing DevOps, for smooth transitioning of product, from the development to the operations team, where they come together to serve the customers.

- **Scalability:** DevOps involves automation of processes and eliminates bottlenecks and inefficiencies across the product lifecycle. This allows organizations to scale more rapidly and propel organizational growth.
- **Reliability:** DevOps implementation involves streamlining the processes which lead to product stability.
- **Optimum utilization of cost and resources:** Similar to the lean practices, DevOps also focuses on maximizing the product value to the customers and eliminating waste at every level.

### 3.1.3 Principles Governing DevOps

Globally organizations, independent bodies and front runners have observed different views and attributed various aspects as the principles

DASA, the DevOps Agile Skills Association, Microsoft, Accenture and other forerunners define DevOps based on principles such as agile, customer-centric action, creating product with end in mind, performance orientation, teamwork, end to end responsibility, cross-functional teams, continuous improvement and automation among many others

DEV	OPS
 <p><b>Application Performance</b> Modern developers use APM tools to decrease latency, have complete visibility into code, databases, caches, queues, and third-party services.</p>	 <p><b>Application Availability</b> The applications need to be up and running and it's Ops responsibility to ensure uptime and SLAs are in order.</p>
 <p><b>End User Analytics</b> A great developer understands end users have the best feedback and analytics play an enormous part of understanding users. Developers are constantly monitoring end user latency and checking performance by devices and browsers.</p>	 <p><b>Application Performance</b> Classic Ops generally rely on infrastructure metrics – CPU, memory, network and disk I/O, etc. Modern Ops correlate all of those metrics with application metrics to solve problems 10x faster</p>
 <p><b>Quality Code</b> Developers need to ensure their deployments and new releases don't implode or degrade the overall performance</p>	 <p><b>End User Complaints</b> The goal is to know about and fix problems before end users complain, reduce the number of support tickets, and eliminate false alerts.</p>
 <p><b>Code-Level Errors</b> When you have a large distributed application its is vital to lower MTTR by finding the root cause of errors and exceptions.</p>	 <p><b>Performance Analytics</b> Automatically generated baselines of all metrics help Ops understand what has changed and where to focus their troubleshooting efforts. Alerts based upon deviation from observed baselines improved alert quality and reduce alert noise.</p>

#### Facilitator Notes:

Introduce the different principles or attributes that organizations across the globe believe, are the governing principles of DevOps.

The different technologies and the forerunners of DevOps have attributed different factors that influence and govern the implementation of DevOps in an organization. Each of the attributes, add another dimension to DevOps and its philosophy.

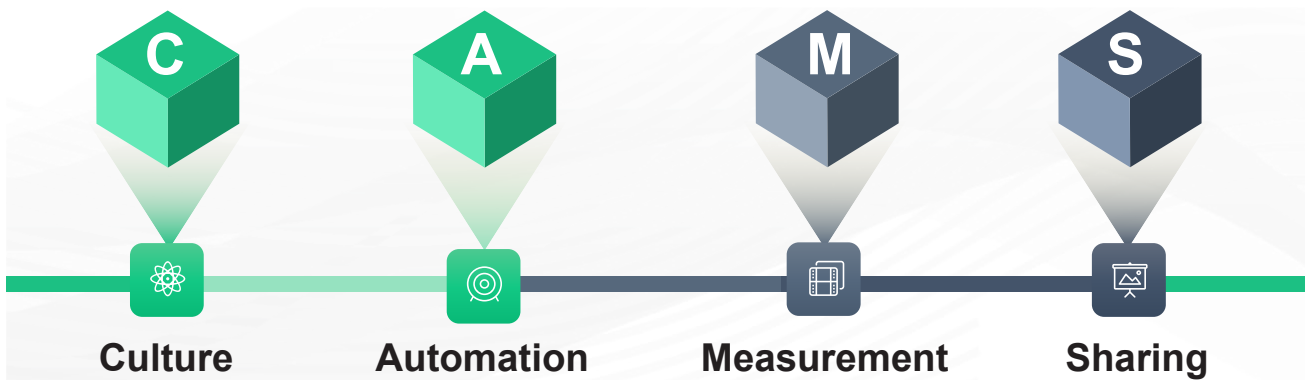
DevOps is predominantly a new age concept that is implemented to expedite the product building and delivery process to ensure that customer needs are met from time to time. Teamwork, cross-functional collaboration, continuous learning and experimentation and automation are all factors that define DevOps.

## 4. CAMS Model – Gene Kim

Gene Kim, the founder of the phoenix project, coined the CAMS model.

CAMS is an acronym that embodies the core principles governing the devops,

- Culture
- Automation
- Measurement
- Sharing



### Facilitator Notes:

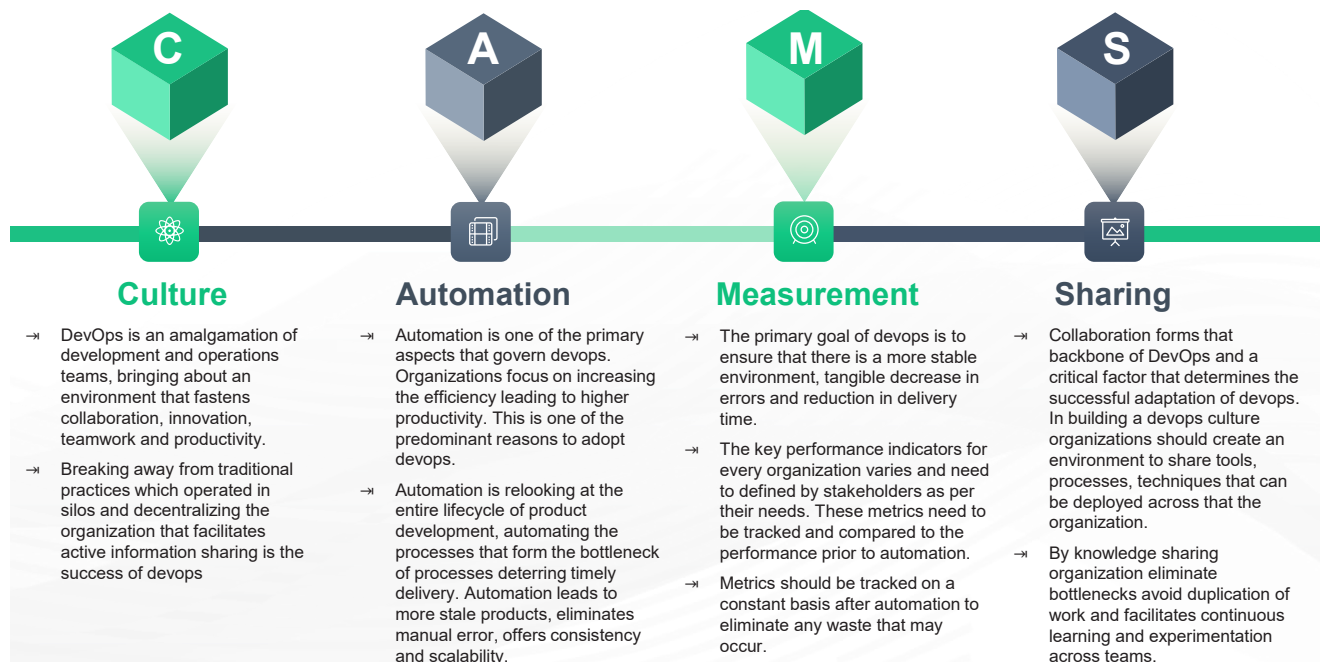
Tell the participants that one of the most important models in DevOps is the 'CAMS model'. It was proposed by 'Gene Kim' and stands for:

- Culture
- Automation
- Measurement
- Sharing

Although different aspects were defined, the one that best fits the definition of DevOps is 'Gene Kim's' CAMS model which is Culture, Automation, Measurement & Sharing.

These four attributes form the crux of DevOps and its philosophy.

## 4.1 Components of the CAMS Model



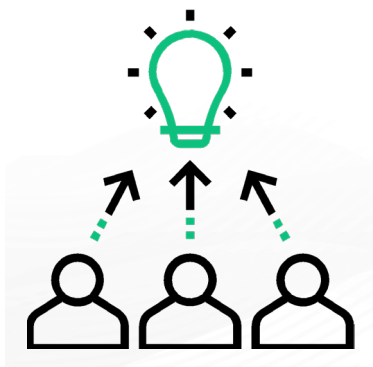
### Facilitator Notes:

Tell the participants that you will go over the components of the CAMS model with them.

Let us deep dive into each of the four components of the CAMS model:

- **Culture:** DevOps propagates a very open culture that promotes trust, eliminates blaming, embraces transparency and allows teams to work and contribute independently. The constantly changing nature requires the organization to be prepared for constant syncing without room for any duplication. DevOps, as a culture should be a fundamental platform on which the organization operates.
- **Automation:** The organizations predominantly look at adapting new processes to expedite the speed of product delivery. Automation helps in removing the backlogs that may occur due to poor processes and replacing it with an efficient automated process.
- **Measurement:** Companies need to constantly track and monitor application performance, number of bugs, recurrence of issues, application downtime etc., to constantly improve the product.
- **Sharing:** A collaborative team environment can only be enabled if there is sharing among individuals, across cross functional teams. Sharing of knowledge, product information, new processes, techniques and anything that could make the environment conducive to speedy delivery of product.

## 4.2 Activity

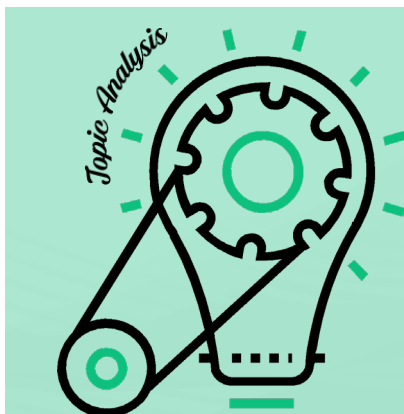


### Facilitator Notes:

- Divide the class into two teams.
- One of the teams represents a Developer and the other represents a System Administrator.
- Ask the teams to choose any 3 applications that they use on a daily basis and write down the different development and administration challenges they could face on a daily basis.
- For example, WhatsApp

Participate in the activity.

## What did you Grasp?



1. Which of the following is not a DevOps principle?
  - A) Culture
  - B) Automation
  - C) Processes
  - D) None of the above
2. Disconnect between development and operations team can occur due to
  - A) Disintegrated processes
  - B) Closed feedback loops
  - C) Lack of collaboration
  - D) Unanimity in usage of tools

### Facilitator Notes:

Tell the participants that it is time for a quick knowledge check.

### Correct Answer:

1. C. Processes
2. A. Disintegrated processes  
B. Closed feedback loops

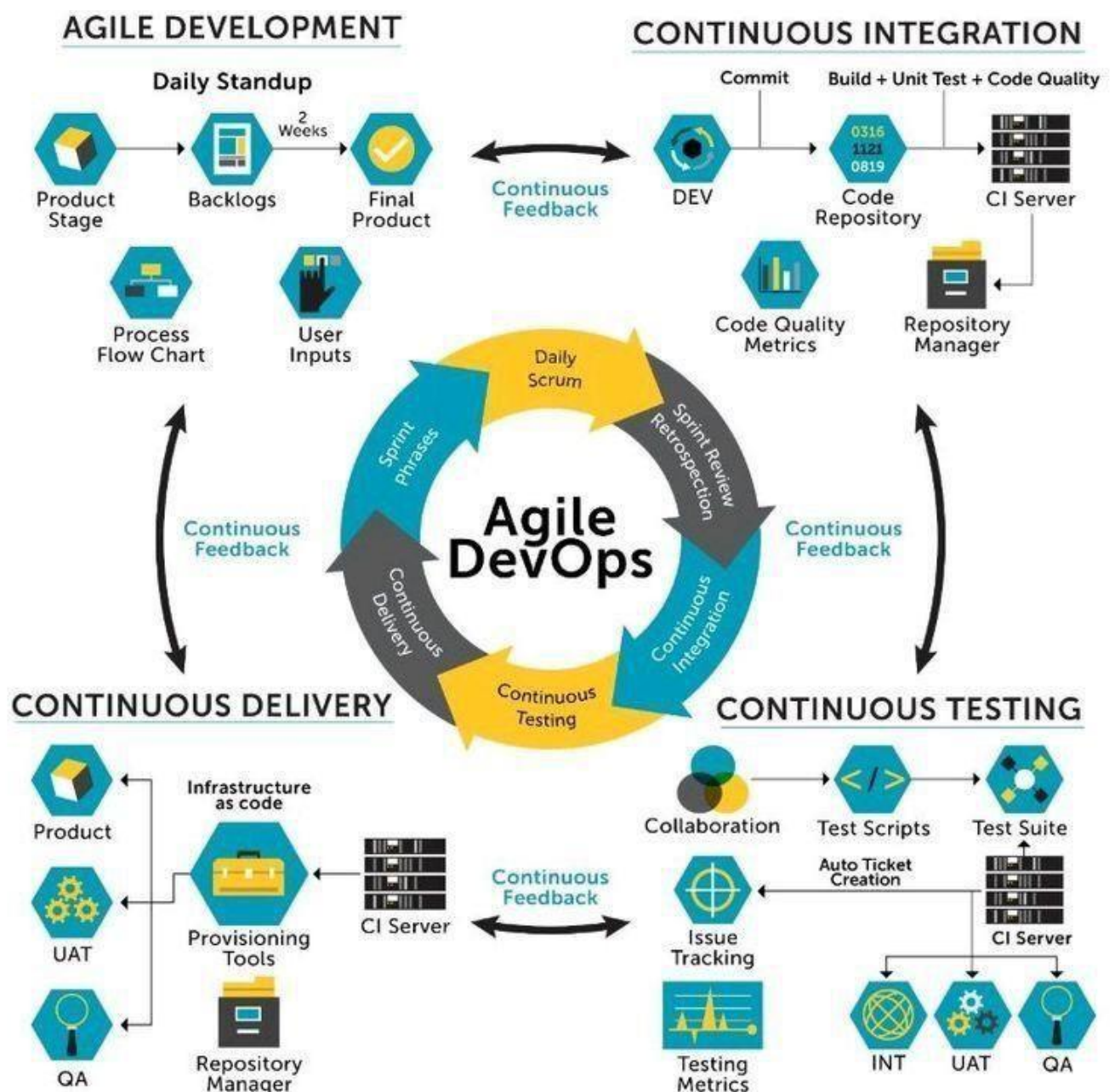
You have to attempt the question.

You have one attempt.

## 5. DevOps and Agile

DevOps and agile are embedded with the same values and principles.

- They increase the efficiency of the product delivery
- They propagate collaboration
- They offer structured processes embracing different functions within an organization
- They help companies in staying relevant





**Facilitator Notes:**

- Tell the participants that DevOps and Agile have the same values and principles.
- Inform the participants that Agile is a part of DevOps.

DevOps and Agile have the same value system and core principles. DevOps is an extension of agile methodologies.

The primary objective of Agile software development is to offer superior value to the customer. It works to integrate development and quality assurance teams for a faster delivery of product.

DevOps is more an extension of agile software development, i.e. DevOps integrates development, quality assurance and operations teams to build products at a faster pace.

Both, DevOps and Agile, propagate the same value of collaboration and active interaction among the teams for a speedy product delivery. Both, Agile and DevOps break the barriers creating cross-functional teams across the organization, to collaborate and create the products that are relevant to the existing customers' need. Both the methodologies follow the same philosophy of catering to ever-growing customers' needs and continuously offering value. Agile and DevOps adopt practices that allow quicker response at every stage of product building.

## 5.1 Agile Methodology

The four values as per the Agile Manifesto are



**Individuals and interactions**



**Working software**



**Customer collaboration**



**Responding to change**





**Facilitator Notes:**

Talk about the values that govern the agile software methodology as discussed in module 2.

The four values are the foundation of the agile software methodology:

- Individuals and interactions over processes and tools
- Functional software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan.

There twelve principles that govern Agile, drafted by industry experts who proposed Agile as an alternative, lightweight methodology to traditional software development methods.

1. Our highest priority is to satisfy the customer through an early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in the development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

## 5.2 Comparison between DevOps and Traditional IT Cultures

Dimension	DevOps Culture	Traditional IT Culture
Batch Size	Micro	Big
Organization	Dedicated cells	Skill-centric silos
Scheduling	Decentralized and Continuous	Centralized
Release	Uneventful	High risk event
Information	Actionable	Disseminated
Culture	Fail early	Do not fail
Metric	Cost, Capacity and Flow	Cost and Capacity
Define "Done"	"It's ready to deploy"	"I did my job"

Source: IBM

### Facilitator Notes:

Explain the participants how DevOps culture and Traditional IT culture compare to each other.

The table summarizes the core differences in adopting DevOps culture and Traditional IT culture. The differences can be understood on the basis of three categories:

### Planning and Organization

- Batch Sizes:** Traditional organizations plan big, have bigger teams to maximize productivity, write huge amounts of code that are bundled and jammed for production. On the other hand, DevOps teams have smaller batch sizes, that are easy to manage and teams can perform frequent product releases.
- Organization:** Skill centric silos form the core of traditional IT culture. Silos have some benefits, but mostly a new feature will go through at least 3-4 silos before reaching the customer. DevOps also has silos but at a different dimension. Teams are arranged in cells, with different cross-functional teams, focused on a single application. Thus a cell is self-sufficient and movement happens smoothly without any hand-offs.
- Scheduling:** Traditional IT culture focuses on efficient scheduling and companies also invest hugely on planning/scheduling systems. These systems are sensitive, but can be inaccurate. In case of DevOps scheduling is handled locally within the cells. Smaller batch sizes, autonomous teams and automated processes aid in efficient planning. Teams will have a better insight, as planning is done only for the near future.

### Performance and Culture

- Release:** Software release phase in traditional IT is an eventful, high-risk operation. The phase is full of issues, escalations and back and forth code movements. The process involves personnel from all levels and management becomes clumsy. In case of DevOps the release phase is almost

uneventful. Code is checked in on a regular basis and testing processes and short feedback loops ensure that the processes are in sync. This makes smoother releases.

- b) **Information:** Information in traditional IT is generated by specialists, which is then combined to form a massive datasets, which again goes for approval and then sent to managers, which reaches the teams finally. Most of the times the report is not given much attention, as it contains massive amounts of data. In DevOps, the team cells generate only the necessary the data that is circulated locally. This gives room for the team to quickly read and work on action items.
- c) **Culture:** Traditional systems oppose failures, by means of stringent processes, approvals, etc. Despite all this, in more than 50% of traditional systems, projects are delivered late. Most of it is due to infrastructure and quality issues. DevOps believes that failure is unavoidable, but failing early helps in faster recovery. Failure at early stages cause only minimal damage, as compared to the final product failure that causes irreparable damage to the business.

## Measure

- a) **Metric:** Cost and capacity are the two measurement models, to see how much gets done and at what cost. DevOps works on three measurement models cost, capacity and flow. Flow forces an organization to take a look at its end to end cycle time, identify areas of waste, calculate true productive time, quantify quality, and focus on activities that add the most value.
- b) **The definition of “Done”:** In traditional IT, ‘Done’ is seen from the individual's viewpoint, and is focused on meeting the hand-off deadline. DevOps is focused on creating dedicating cross-functional teams. It's the goal of the team: Bring the software to market.

## 5.3 Why to Build a Business Case for DevOps

Here are the 10 reasons to build a business case for Devops

- 1 Collaboration
- 2 Improved speed to market
- 3 No silos, no waste
- 4 Encouraging innovation and creativity
- 5 Effective utilization of resources and reduction in cost
- 6 Increased employee engagement and job satisfaction
- 7 Continuous integration and delivery
- 8 Fewer Failures
- 9 Increased Performance
- 10 Stability

**Facilitator Notes:**

Explain the participants about the need for building a business case for DevOps.

There are 10 major reasons why we should build a business case for DevOps. Let's see each one of them in detail here.

**1. Collaboration**

For effective problem solving teams need to come together and use the available time and resources. Because, challenges do not affect the individual teams alone. It is the business that gets affected and regardless of who is responsible, the problem needs to be solved. Without collaboration, this process takes longer and can create further problems that may not be immediately apparent. DevOps fosters collaboration, hence efficient problem solving. Togetherness and communication between the teams help to work faster and smarter and similar issues can be prevented.

**2. Improved speed to market**

Improved speed to market is critical for organizations to get the early mover advantage. Businesses need to gain a competitive edge in an industry where software and tools are outmoded almost as quickly as they are released. Introducing a DevOps approach will enable an organization to go from an initial concept to a viable product in a shorter timescale.

**3. No silos, no waste**

By combining multiple teams and disciplines into one lean, mean DevOps team that has cross-functional skill sets and communicate efficiently, DevOps allows teams to complete tasks quickly and efficiently while maintaining stability and quality.

**4. Encouraging innovation and creativity**

Continuous Integration, standardized production environments, and automated deployments allow practitioners to focus on the more inventive and creative side of their role. The environment and culture of DevOps encourage a deeper understanding and implementation of best practices in an organization.

**5. Effective utilization of resources and reduction in cost**

By implementing a DevOps approach, the costs and demand for resources associated with traditional IT implementations is significantly reduced. When the organizations use continuous delivery and Lean Management practices, higher quality results and shorter cycle times are achieved which further reduce costs. Other factors that help to reduce cost and resource requirements, include minimal project start-up and ongoing operational costs, increased collaboration, increased data availability and accessibility, and improved security.

## 6. Increased employee engagement and job satisfaction

DevOps provides a collaborative and multi-skilled environment which contributes heavily to job satisfaction. DevOps practices and culture increase employee satisfaction which leads to better business outcomes.

## 7. Continuous integration and delivery

Continuous integration is a development practice that involves deploying code to a shared repository several times per day. Using an automated build process in combination with automated testing helps to verify each check-in, which produces more stable software.

## 8. Fewer Failures

The 2014 State of DevOps report showed that high-performing organizations had 50 percent fewer failures. The 2015 State of DevOps report showed a continuation in trend by revealing that the organizations who adopt a DevOps mindset and culture have 60 times fewer failures than those not implementing a DevOps approach.

## 9. 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, hence, leading to increased performance.

## 10. Stability

DevOps allows a single team to handle both, new functionality and the stability of the system. Each team member takes ownership of the business goals. Deploying often and in smaller, indivisible groups allows engineers to troubleshoot and resolve issues faster. The combination of tools and best practices, along with automation allows a DevOps team to increase overall stability.

## What did you Grasp?



1. The following is not a value of Agile software methodology?
  - A) Interaction & individuals
  - B) Collaboration
  - C) Sequential development approach
  - D) Quicker response time
2. The communication channels in agile and devops teams have to be linear and follow a sequential order.
  - A) True
  - B) False



3. Agile and devops methodologies can be successful only if, \_\_\_\_\_ exists
- A) Open collaboration
  - B) Disintegration
  - C) Different processes
  - D) Tools & technologies

### Facilitator Notes:

Tell the participants that it is time for a quick knowledge check.

#### Correct Answer

- 1. C. Sequential development approach
- 2. B. False
- 3. A. Open collaboration

### In a nutshell, we learnt:



- 1. Traditional IT systems, its pitfalls and how this led to the emergence of DevOps philosophy
- 2. Jene Kim's CAMS model and the important principles that govern DevOps
- 3. The Agile methods, its values and 12 principles as per the agile manifesto
- 4. The similarities between agile and devops methodology

**Facilitator Notes:**

Give a brief summary of all the different topics discussed during this module.

We come to the end of this module. Here, we learnt about the drawbacks of traditional IT systems and how it is not suitable for new-age applications. The emergence of DevOps and its principles. The CAMS model of DevOps and Agile methodology and how DevOps is an extension of Agile methodology.





# Release Notes

## B. TECH CSE with Specialization in DevOps Semester One -Year 01

### Release Components.

Facilitator Guide, Facilitator Course Presentations, Student Guide and Mock exams.

### Current Release Version.

1.0.0

### Current Release Date.

2 July 2018

### Course Description.

Xebia, has been recognized as a leader in DevOps by Gartner and Forrester and this course is created by Xebia to equip students with set of practices, methodologies and tools that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

### Copyright © 2018 Xebia. All rights reserved.

Please note that the information contained in this classroom material is subject to change without notice. Furthermore, this material contains proprietary information that is protected by copyright. No part of this material may be photocopied, reproduced, or translated to another language without the prior consent of Xebia or ODW Inc. Any such complaints can be raised at [sales@odw.rocks](mailto:sales@odw.rocks)

The language used in this course is US English. Our sources of reference for grammar, syntax, and mechanics are from The Chicago Manual of Style, The American Heritage Dictionary, and the Microsoft Manual of Style for Technical Publications.

<b>Bugs reported</b>	Not applicable for version 1.0.0
<b>Next planned release</b>	Version 2.0.0 Feb 2019