Introduction

Completed

- 4 minutes

"DevOps is the union of people, process, and products to enable continuous delivery of value to our end users." - According to Donovan Brown in What is DevOps?

The DevOps learning paths will help you prepare for a DevOps journey. You'll learn the main characteristics of the DevOps process, tools, and people involved during the lifecycle. Also, it prepares you for the Microsoft DevOps Solution certification exam. You'll see other content to ensure you have a complete picture of DevOps. The module's content includes graphics, reference links, module review questions, and optional hands-on labs.

You'll learn the following:

- How to plan for DevOps.
- Use source control.
- Scale Git for an enterprise.
- Combine artifacts.
- Design a dependency management strategy.
- Manage secrets.
- Implement continuous integration.
- Implement a container-build strategy.
- Design a release strategy.
- Set up a release management workflow.
- Implement a deployment pattern.
- Optimize feedback mechanisms.

Plan before you act. This module will help you understand what DevOps is and how to plan for a DevOps transformation journey.

What is the DevOps transformation journey?

The DevOps transformation journey is a series of 9 learning paths. It familiarizes you with Azure DevOps and GitHub. Also, learn its many services, features, and integration with tools to support your DevOps process.

Why should I take the DevOps learning path?

People in these modules are interested in designing and implementing DevOps processes. Also, they're preparing for the AZ-400 - Designing and Implementing Microsoft DevOps Solutions certification exam.

The certification exam is for DevOps professionals. Combine people, processes, and technologies to continuously deliver valuable products and services that meet end-user needs and business goals. DevOps professionals streamline delivery by optimizing practices, improving communications and collaboration, and creating automation.

They design and implement application code and infrastructure strategies that allow continuous integration, testing, delivery, monitoring, and feedback.

Exam candidates must be proficient with Agile practices. They must be familiar with Azure administration, development and experts in at least one of these areas.

DevOps professionals must design and implement DevOps practices for version control, compliance, infrastructure as code, configuration management, build, release, and testing-using Azure technologies.

There are five domain areas.

AZ-400 Domain Area	Weight
Configure Processes and Communications.	13%
Design and Implement Source Control.	19%
Design and Implement Build and Release Pipelines.	42%

AZ-400 Domain Area	Weight
Develop a Security and Compliance Plan.	14%
Implement an Instrumentation Strategy.	13%

Learning objectives

After completing this module, students and professionals can:

- Plan for the transformation with shared goals and timelines.
- Select a project and identify project metrics and Key Performance Indicators (KPIs).
- Create a team and agile organizational structure.
- Design a tool integration strategy.
- Design a license management strategy (for example, Azure DevOps and GitHub users).
- Design a plan for end-to-end traceability from work items to working software.
- Design an authentication and access strategy.
- Design a strategy for integrating on-premises and cloud resources.

Prerequisites

Successful learners will have prior knowledge and understanding of the following:

- Cloud computing concepts include understanding PaaS, SaaS, and IaaS implementations.
- Azure administration and Azure development with proven expertise in at least one of these areas.
- Version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

If you're new to Azure and cloud computing, consider one of the following resources:

- Free online: Azure Fundamentals .
- Instructor-led course: AZ-900: Azure Fundamentals .

If you're new to Azure Administration, consider taking the:

- Free online: Prerequisites for Azure Administrators .
- Instructor-led courses: AZ-104: Microsoft Azure Administrator .

If you're new to Azure Developer, consider taking the:

- Free online: Create serverless applications .
- Instructor-led courses: AZ-204: Developing Solutions for Microsoft Azure and AZ-020: Microsoft Azure Solutions for AWS Developers .

You must create an Azure DevOps Organization and a Team Project for some exercises. If you don't have it yet, see the following:

- Create an organization - Azure DevOps .
- If you already have your organization created, use the Azure DevOps Demo Generator [https://azuredevopsdemogenerator.azurewebsites.net] and create a new Team Project called "Parts Unlimited" using the template "PartsUnlimited." Or feel free to create a blank project. See Create a project - Azure DevOps .

You must create a GitHub account at GitHub.com and a project for some exercises. If you don't have it yet, see the following:

- Join GitHub · GitHub
- If you already have your GitHub account, create a new repository Creating a new repository - GitHub Docs .

Continue

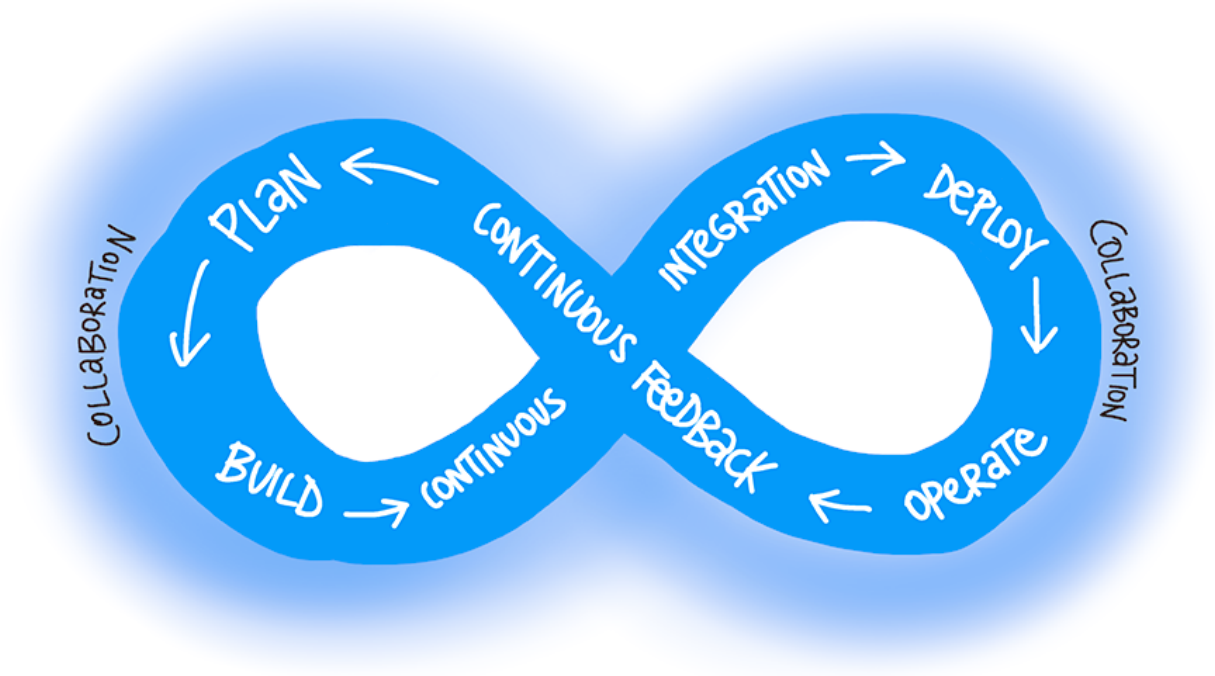
Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

What is DevOps?

Completed

- 3 minutes

The contraction of "Dev" and "Ops" refers to replacing siloed Development and Operations. The idea is to create multidisciplinary teams that now work together with shared and efficient practices and tools. Essential DevOps practices include agile planning, continuous integration, continuous delivery, and monitoring of applications. DevOps is a constant journey.



Understand your cycle time

Let us start with a basic assumption about software development. We will describe it with the OODA (Observe, Orient, Decide, Act) loop. Originally designed to keep fighter pilots from being shot out of the sky, the OODA loop is an excellent way to think about staying ahead of your competitors. You start with observing business, market, needs, current user behavior, and

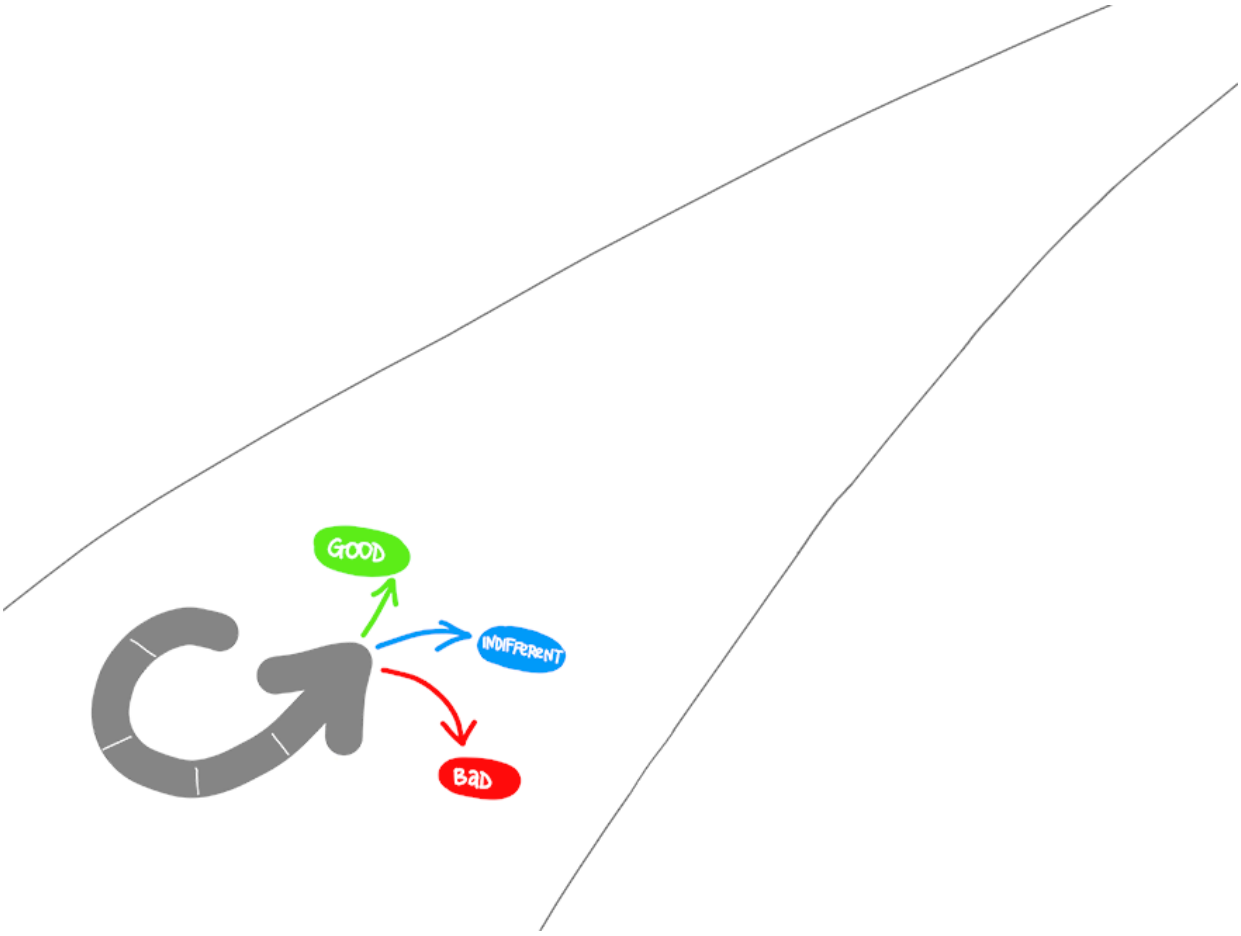
available telemetry data. Then you orient with the enumeration of options for what you can deliver, perhaps with experiments. Next, you decide what to pursue, and you act by delivering working software to real users. You can see all occurring in some cycle time.

Become data-informed

We recommend you use data to inform what to do in your next cycle. Many experience reports tell us that roughly one-third of the deployments will have negative business results. Approximately one-third will have positive results, and one-third will make no difference. Fail fast on effects that do not advance the business and double down on outcomes that support the business. Sometimes the approach is called pivot or persevere.

Strive for validated learning

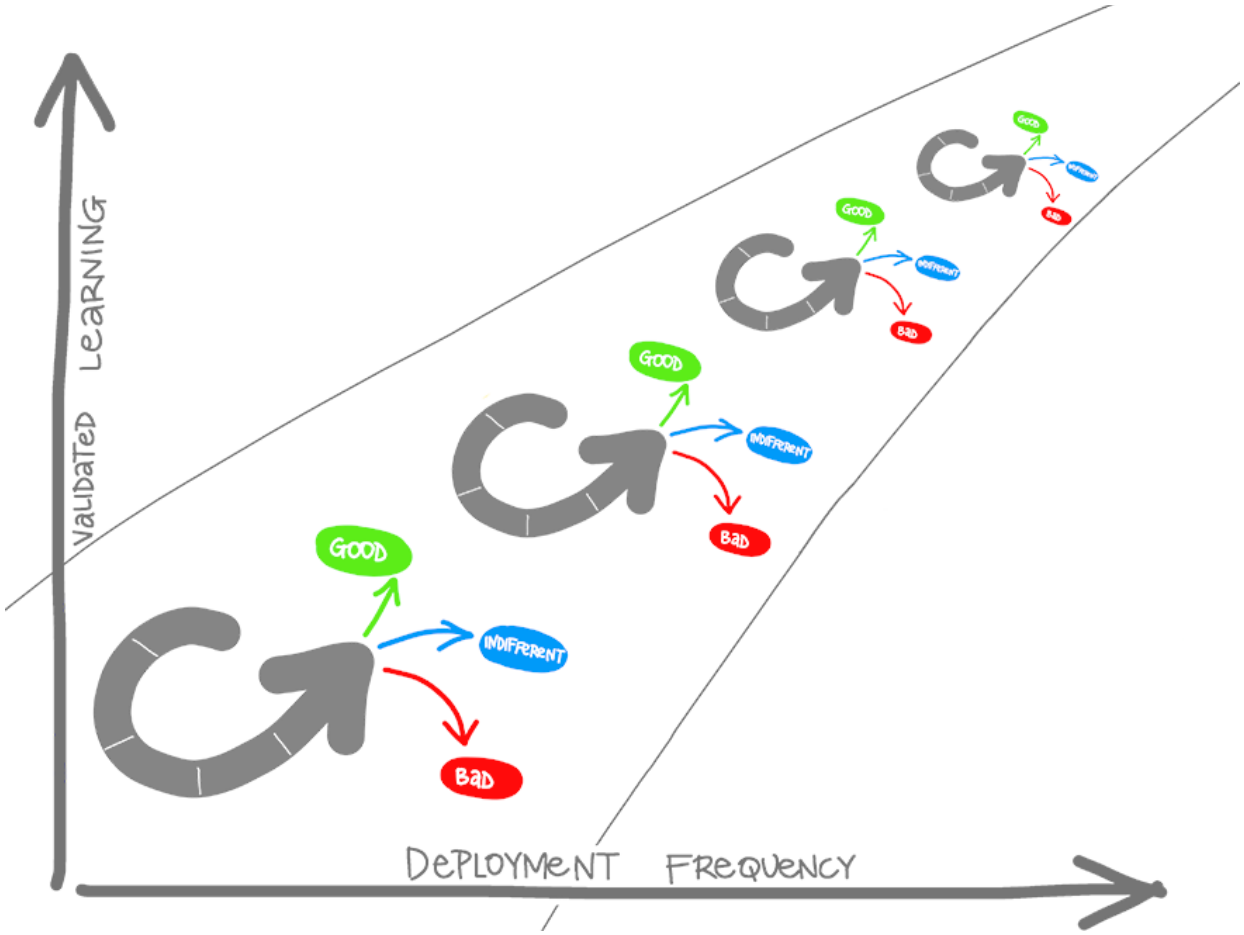
How quickly you can fail fast or double down is determined by your cycle time. Also, in how long that loop takes, or in lean terms. Your cycle time determines how quickly you can gather feedback to determine what happens in the next loop. The feedback that you collect with each cycle should be factual, actionable data. We call it validated learning.



Shorten your cycle time

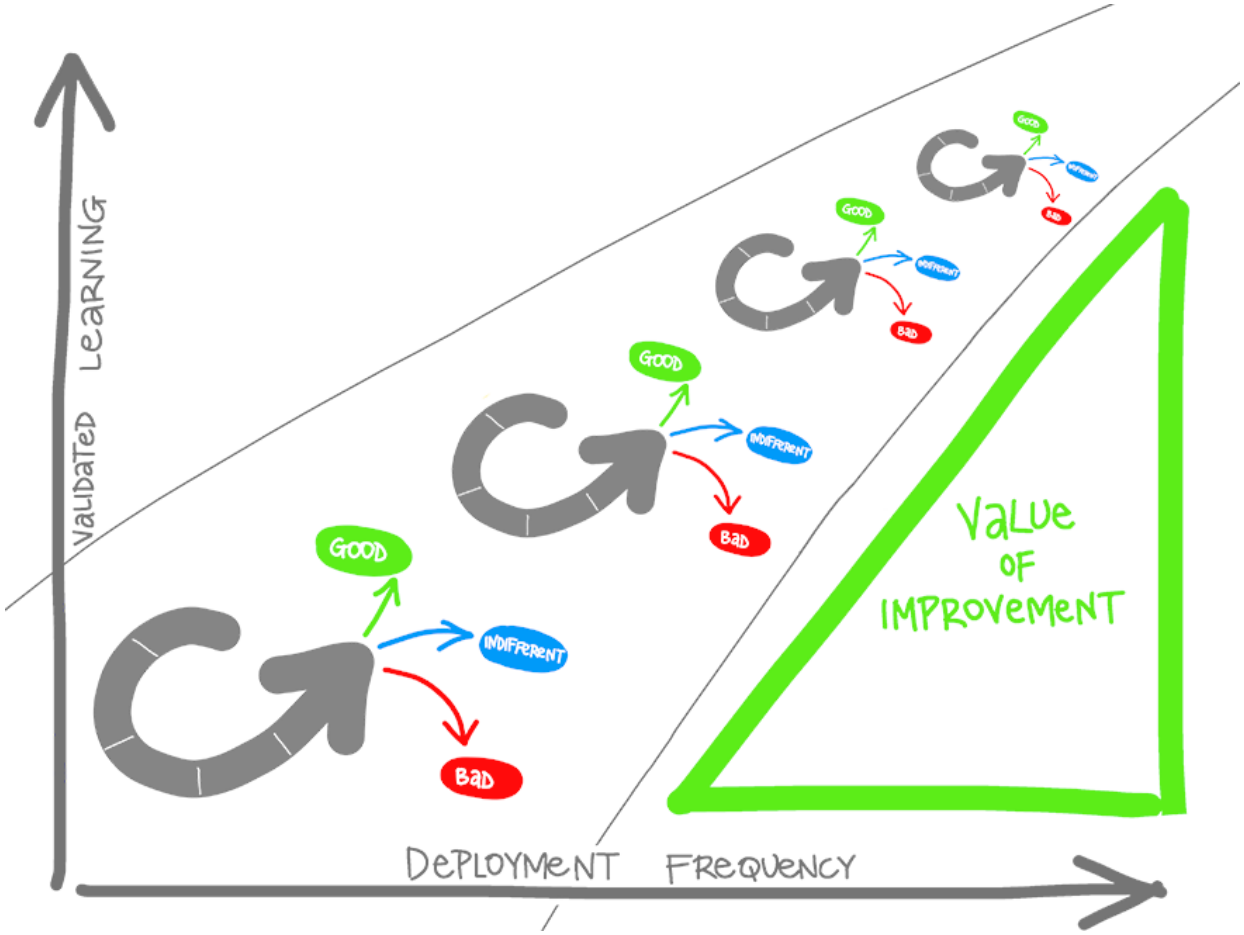
When you adopt DevOps practices:

- You shorten your cycle time by working in smaller batches.
- Using more automation.
- Hardening your release pipeline.
- Improving your telemetry.
- Deploying more frequently.



Optimize validated learning

The more frequently you deploy, the more you can experiment. The more opportunity you have to pivot or persevere and gain validated learning each cycle. This acceleration in validated learning is the value of the improvement. Think of it as the sum of progress that you achieve and the failures that you avoid.



Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

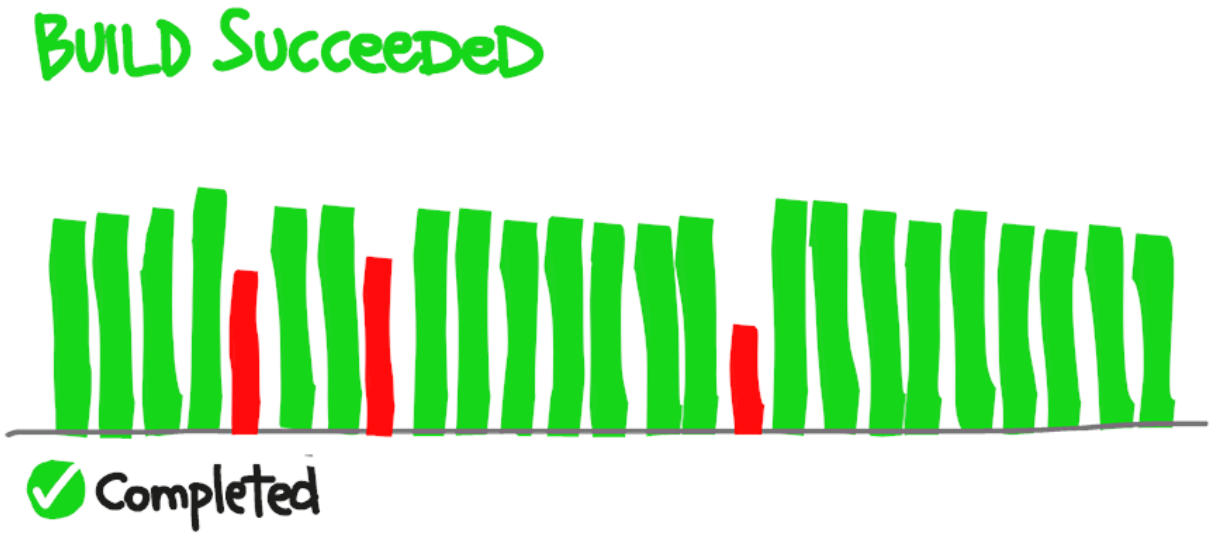
Explore the DevOps journey

Completed

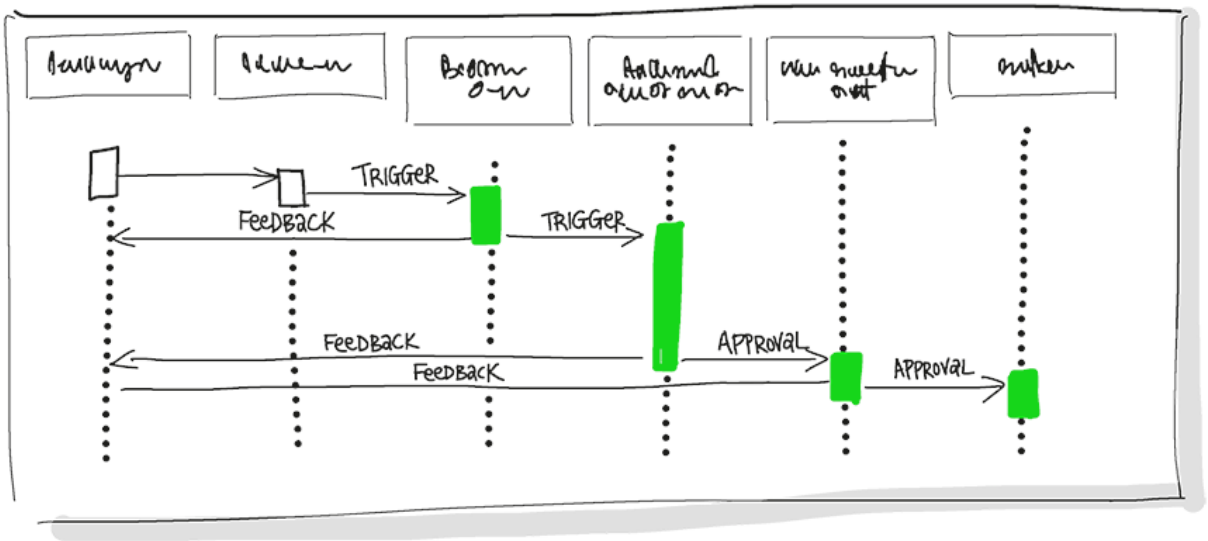
- 4 minutes

Remember, the goal is to shorten cycle time. Start with the release pipeline. How long does it take to deploy a change of one line of code or configuration? Ultimately, that is the brake on your velocity.

- Continuous Integration drives the ongoing merging and testing of code, leading to an early finding of defects. Other benefits include less time wasted fighting merge issues and rapid feedback for development teams.

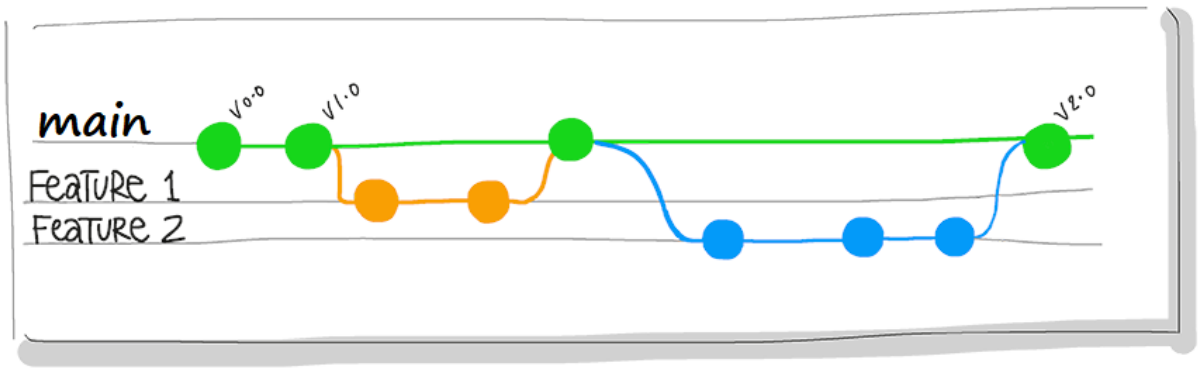


- Continuous Delivery of software solutions to production and testing environments helps organizations quickly fix bugs and respond to ever-changing business requirements.

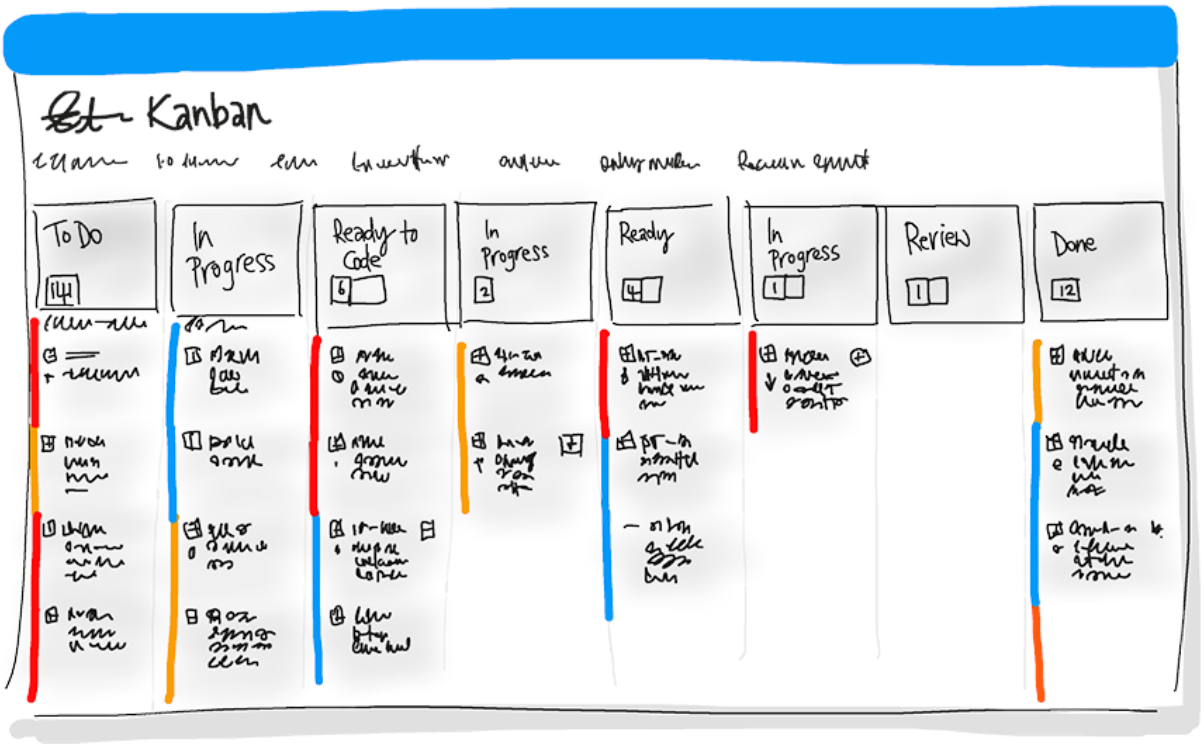


- Version Control, usually with a Git-based Repository, enables teams worldwide to communicate effectively during daily development

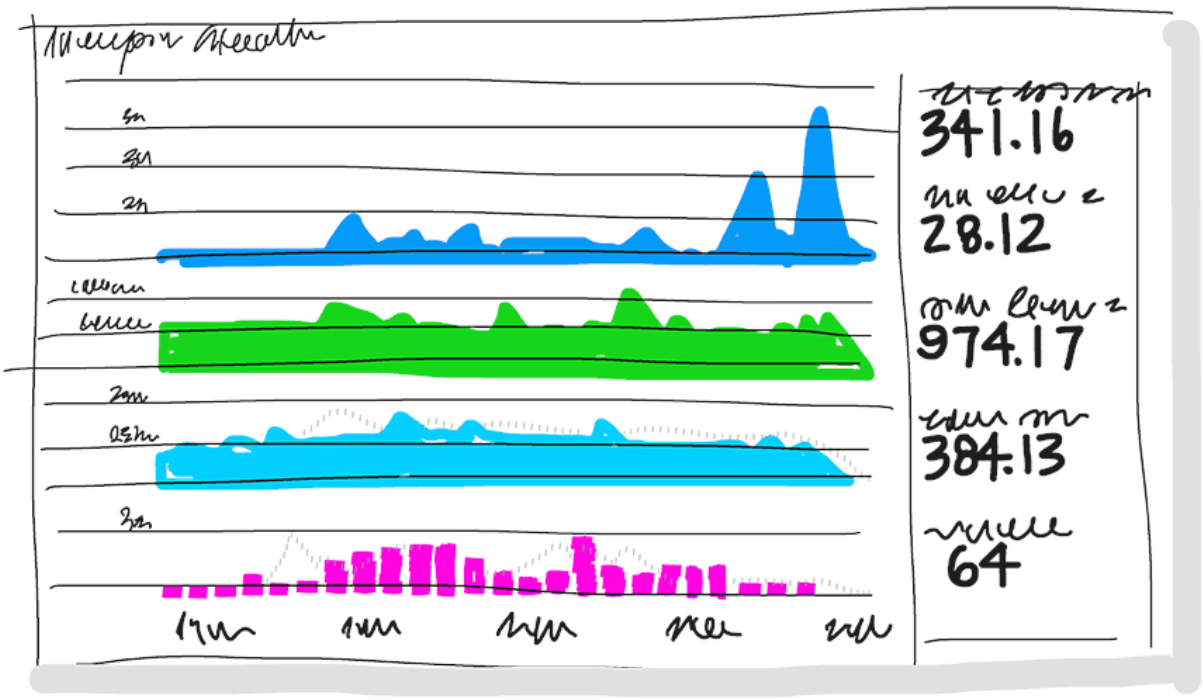
activities. Also, integrate with software development tools for monitoring activities such as deployments.



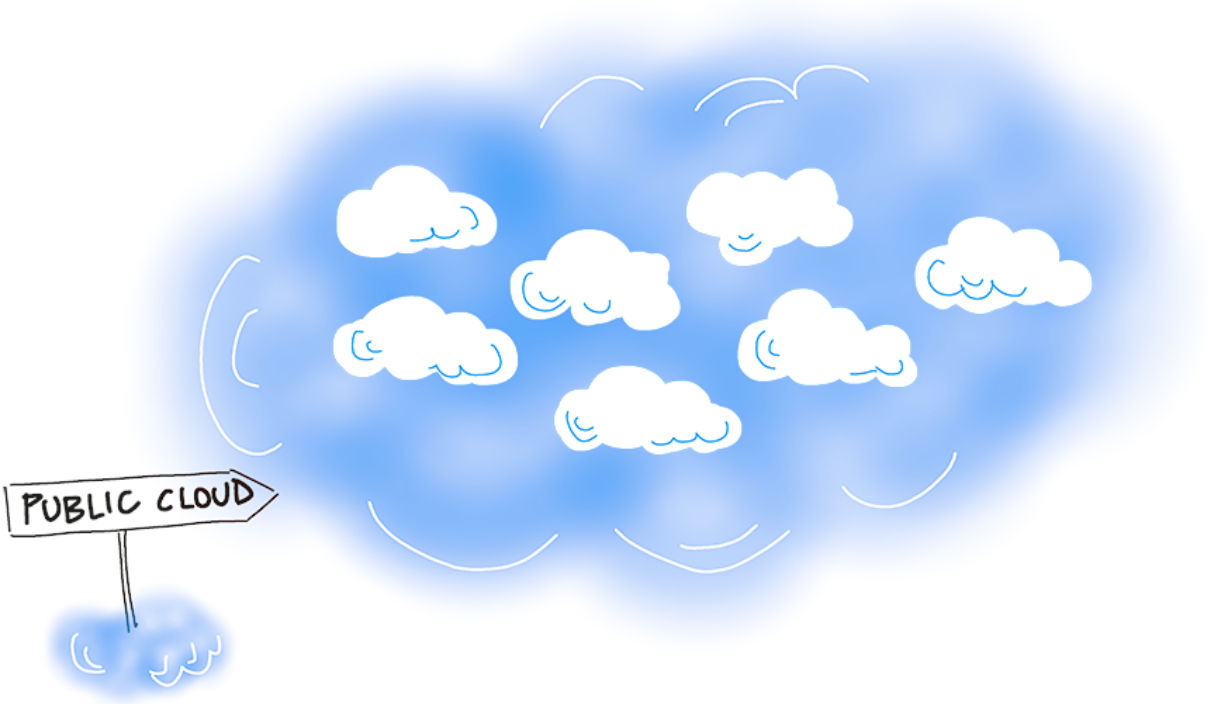
- Use Agile planning and lean project management techniques to:
  - Plan and isolate work into sprints.
  - Manage team capacity and help teams quickly adapt to changing business needs.
  - A DevOps Definition of Done is working software collecting telemetry against the intended business goals.



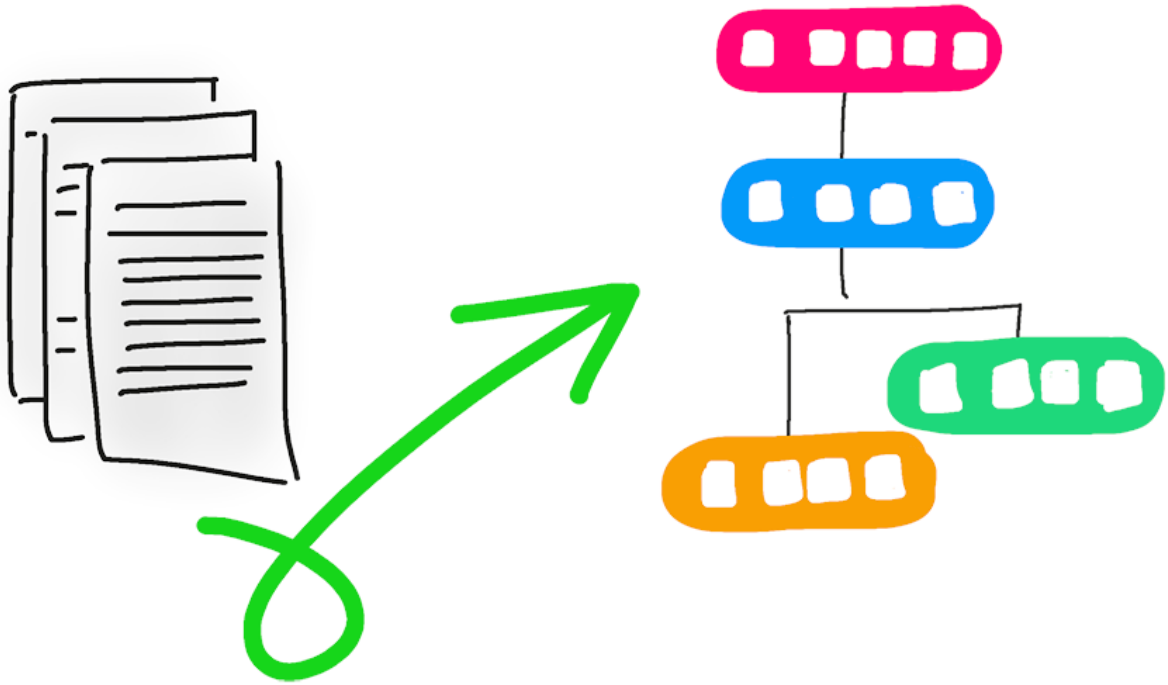
- Monitoring and Logging of running applications. Including production environments for application health and customer usage. It helps organizations create a hypothesis and quickly validate or disprove strategies. Rich data is captured and stored in various logging formats.



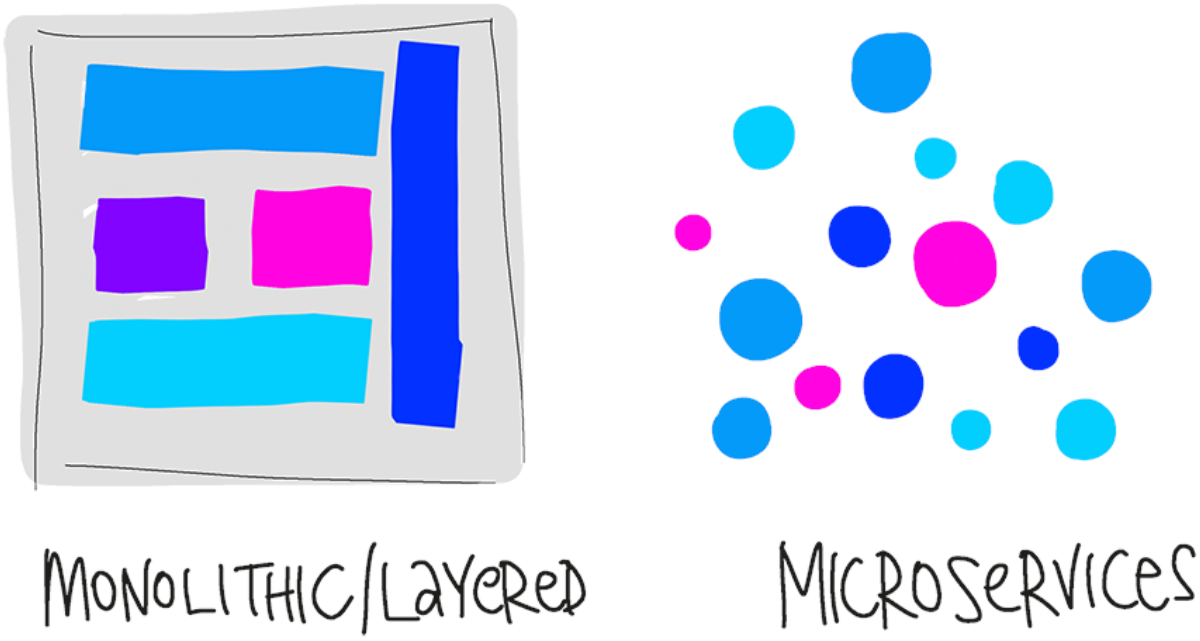
- Public and Hybrid Clouds have made the impossible easy. The cloud has removed traditional bottlenecks and helped commoditize Infrastructure. You can use Infrastructure as a Service (IaaS) to lift and shift your existing apps or Platform as a Service (PaaS) to gain unprecedented productivity. The cloud gives you a data center without limits.



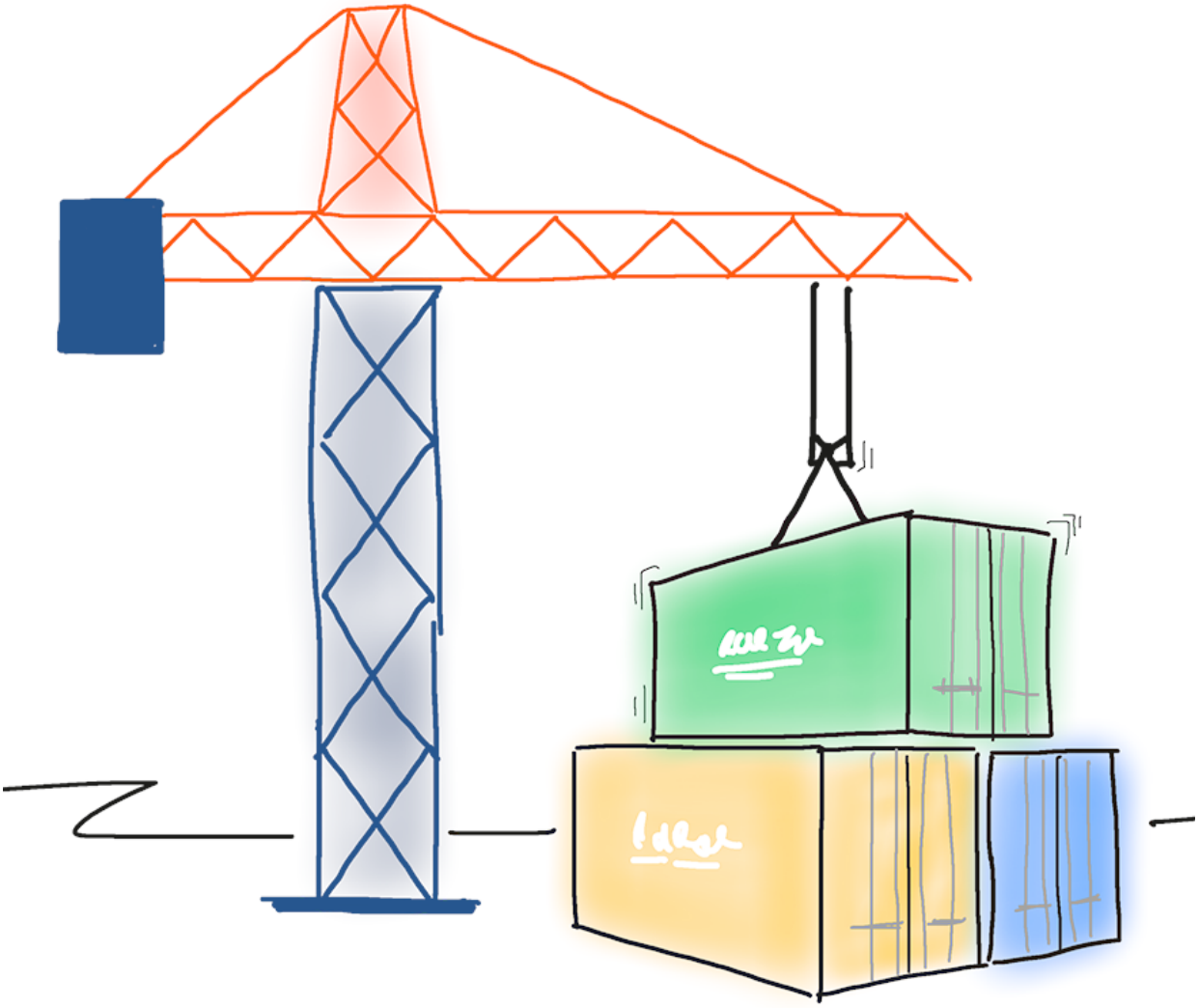
- Infrastructure as Code (IaC): Enables the automation and validation of the creation and teardown of environments to help deliver secure and stable application hosting platforms.



- Use Microservices architecture to isolate business use cases into small reusable services that communicate via interface contracts. This architecture enables scalability and efficiency.



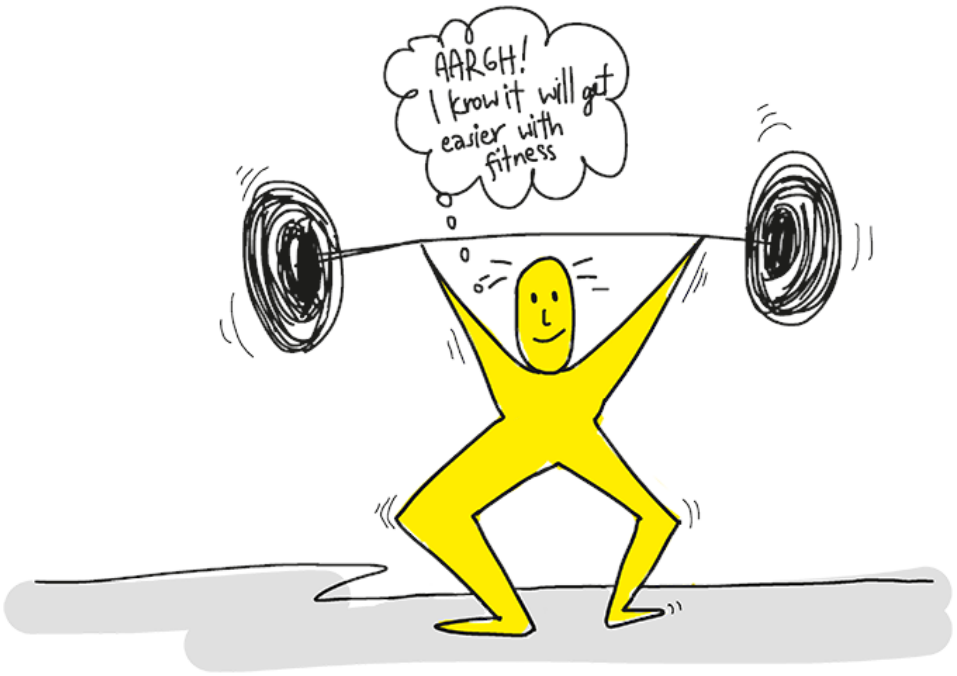
- Containers are the next evolution in virtualization. They're much more lightweight than virtual machines, allow much faster hydration, and easily configure files.



DevOps may hurt at first.

If it hurts, do it more often. Adopting new practices like going to the gym is likely to hurt first. The more you exercise the new techniques, the easier they'll become.

Like training at the gym, where you first exercise large muscles before small muscles, adopt practices that have the most significant impact. Cross-train to develop synergy.



Note

The source article defines DevOps .

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Identify transformation teams

Completed

- 2 minutes

Unless you're building an entirely new organization, one of the significant challenges with any DevOps Transformation Project is taking actions that conflict. At least in some way with ongoing business states.

The first challenge is the availability of staff. Suppose the staff members leading the transformation project are also involved in existing day-to-day work within the organization.

It will be challenging to focus on the transformation when their current role directly impacts customer outcomes.

We all know desperate situations involving customers will always win over a long-term project, like DevOps Transformations.

Another issue will be how the organization operates—implementing existing processes and procedures to support current business outcomes.

The disruption required for a true DevOps Transformation will usually challenge existing processes and procedures. Doing that is often difficult.

In the book "Beyond the Idea: How to Execute Innovation," Dr. Vijay Govindarajan and Dr. Chris Trimble noted when successful, they have researched what is involved in allowing Innovation to occur in organizations.

It has often been despite the existing organizational processes. Concluding it only works by creating a separate team to pursue the transformation.

For DevOps transformations, the separate team should be composed of staff members. Focused on and measured the transformation outcomes and not involved in the day-to-day operational work. The team might also include external experts that can fill the knowledge gaps—also helping to advise on processes that are new to the existing staff members.

Ideally, the staff members recruited should already be well-regarded throughout the organization. They should offer a broad knowledge base to think outside the box as a group.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Explore shared goals and define timelines

Completed

- 2 minutes

Explore shared goals

These outcomes should include specific, measurable targets like:

- Reduce the time spent on fixing bugs by 60%.
- Reduce the time spent on unplanned work by 70%.
- Reduce the out-of-hours work required by staff to no more than 10% of total working time.
- Remove all direct patching of production systems.

Note

One of the aims of DevOps is to provide more excellent customer value, so outcomes should have a customer value focus.

Define timelines for goals

Measurable goals also need to have timelines. While it is easy to set longer-term goals, it is also easy to put off work when you do not require it for a while.

It is essential to have an ongoing series of short-term goals. Overall, projects should have timelines that span anywhere from a few months to a year or two in any DevOps transformation project.

Every few weeks, the improvements should be clear and measurable. Ideally, evident to the organization or its customers.

The timeline should not be too short and should always be challenging yet achievable. A review should occur after each short-term goal to help plan the next one.

There are several advantages of the shorter timelines:

- It is easier to change plans or priorities when necessary.
- The reduced delay between doing work and getting feedback helps ensure that the learnings and feedback are incorporated quickly.
- It is easier to keep organizational support when positive outcomes are clear.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Knowledge check

Completed

- 4 minutes

Choose the best response for each question. Then select **Check your answers** .

Check your knowledge

1.

Which of the following choices best describes DevOps?

○

DevOps is the role of who manages source control, pipelines, and monitor environments to continue delivering value to the software project.

○

DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.

○

DevOps is the new process of creating continuous delivery and continuous integration for software projects.

2.

Which of the following choices drives the ongoing merging and testing of code that leads to finding defects early?

○

Continuous Integration.

○

Continuous Delivery.

○

Continuous Feedback.

3.

Which of the following choices is a practice that enables the automated creation of environments?

○

Infrastructure as a Service (IaaS).

Infrastructure as Code (IaC).

Software as a Service (SaaS).

Check your answers

You must answer all questions before checking your work.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Summary

Completed

- 2 minutes

This module explored the key areas that organizations must apply to start their DevOps transformation Journey, changing the team's mindset, and defining timelines and goals.

You learned how to describe the benefits and usage of:

- Understand what DevOps is and the steps to accomplish it.
- Identify teams to implement the process.
- Plan for the transformation with shared goals and timelines.
- Plan and define timelines for goals.

Learn more

- Donovan Brown | What is DevOps?
- What is DevOps? - Azure DevOps | Microsoft Docs
- Getting started with GitHub - GitHub Docs
- View of features and epics on the Feature Timeline - Azure DevOps | Microsoft Docs
- Plan and track work in Azure Boards with Basic or Agile processes - Azure Boards | Microsoft Docs
- Agile Manifesto for Software Development | Agile Alliance
- 12 Principles Behind the Agile Manifesto | Agile Alliance

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .