Introduction

Completed

- 2 minutes

Are you interested in implementing DevOps practices but are not sure what it can do for you?

This module helps organizations decide how to apply the DevOps process and tools to minimize initial resistance.

Learning objectives

After completing this module, students and professionals can:

- Understand different projects and systems to guide the journey.
- Select a project to start the DevOps transformation.
- Identify groups to minimize initial resistance.
- Identify project metrics and Key Performance Indicators (KPIs).

Prerequisites

- Understanding of what DevOps is and its concepts.

Continue

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

Explore greenfield and brownfield projects

Completed

- 2 minutes

The terms greenfield and brownfield have their origins in residential and industrial building projects.

A greenfield project is one done on a green field, undeveloped land. A brownfield project is done on the used ground for other purposes.

Because of the land use that has once occurred, there could be challenges reusing the land. Like existing buildings, some would be obvious but less obvious, like polluted soil.

Applied to software or DevOps Projects

The same terms are routinely applied to software projects and commonly describe DevOps Projects.

On the surface, it can seem that a greenfield DevOps project would be easier to manage and to achieve success.

- There was no existing codebase.
- No existing team dynamics of politics. Possibly no current, rigid processes.

A common misconception is that DevOps is only for greenfield projects and suits startups best. However, DevOps can also succeed with brownfield projects.

The beauty of these projects is that there's often a large gap between customer expectations and delivery.

The teams involved may well realize that the status quo needs to change. They've lived the challenges and the limitations associated with what they're currently doing.

Continue

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

Decide when to use greenfield and brownfield projects

Completed

- 3 minutes

When starting a DevOps transformation, you might need to choose between greenfield and brownfield projects.

There's a common misconception that we need to demystify that DevOps suits greenfield projects better than brownfield projects.

Greenfield projects

A greenfield project will always appear to be a more accessible starting point. A blank slate offers the chance to implement everything the way that you want.

You might also have a better chance of avoiding existing business processes that don't align with your project plans.

Suppose current IT policies don't allow the use of cloud-based infrastructure. In that case, the project might be qualified for entirely new applications designed for that environment from scratch.

For example, you can sidestep internal political issues that are well entrenched.

Brownfield projects

Usually, brownfield projects come with:

- The baggage of existing codebases.
- Existing teams.
- A significant amount of technical debt.

But, they can still be ideal projects for DevOps transformations.

When your teams spend large percentages of their time just maintaining existing brownfield applications, you have limited ability to work on new code.

It's essential to find a way to reduce that time and to make software release less risky. A DevOps transformation can provide that.

The limitations will have often worn down the existing team members. For example, they're working in the past and aren't keen to experiment with new ideas.

The system is often crucial for organizations. It might also be easier to gain more robust management buy-in for these projects because of the potential benefits delivered.

Management might also have a stronger sense of urgency to point brownfield projects in an appropriate direction when compared to greenfield projects that don't currently exist.

Take the first step

Eventually, the goal will be to evolve your entire organization. In looking to take the first step, many organizations start with a greenfield project and then move on.

Continue

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

Decide when to use systems of record versus systems of engagement

Completed

- 2 minutes

When selecting systems as candidates for starting a DevOps transformation, it is necessary to consider the types of systems that you operate.

Some researchers suggest that organizations often use Bimodal IT, a practice of managing two separate, coherent modes of IT delivery - one focused on stability and predictability and the other on agility.

Systems of record

Systems that provide the truth about data elements are often-called systems of record. These systems have historically evolved slowly and carefully. For example, it is crucial that a banking system accurately reflects your bank balance. Systems of record emphasize accuracy and security.

Systems of engagement

Many organizations have other systems that are more exploratory. These often use experimentation to solve new problems. Systems of engagement are modified regularly. Usually, it is a priority to make quick changes over ensuring that the changes are correct.

There is a perception that DevOps suits systems of engagement more than systems of record. The lessons from high-performing companies show that is not the case.

Sometimes, the criticality of doing things right with a system of record is an excuse for not implementing DevOps practices.

Worse, given the way that applications are interconnected, an issue in a system of engagement might end up causing a problem in a system of record anyway.

Both types of systems are great. At the same time, it might be easier to start with a system of engagement when first beginning a DevOps Transformation.

DevOps practices apply to both types of systems. The most significant outcomes often come from transforming systems of record.

Continue

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

Identify groups to minimize initial resistance

Completed

- 2 minutes

Not all staff members within an organization will be receptive to the change required for a DevOps transformation.

In discussions around continuous delivery, we usually categorize users into three general buckets:

- **Canary** users voluntarily test bleeding edge features as soon as they're available.
- **Early adopters** who voluntarily preview releases, considered more refined than the code that exposes canary users.
- **Users** who consume the products after passing through canary and early adopters.

It's essential to find staff members keen to see new features as soon as they're available and highly tolerant of issues when choosing Canary.

Early adopters have similar characteristics to the Canaries. They often have work requirements that make them less tolerant of issues and interruptions to work.

While development and IT operations staff might generally be less conservative than users.

The staff will also range from traditional to early adopters, and others happy to work at the innovative edge.

Ideal target improvements

It's also important to roll out changes incrementally. There is an old saying in the industry that any successful large IT system was previously a successful small IT system.

Large-scale systems rolled out all at once have an abysmal record of success. Most fail, no matter how much support management has provided.

When starting, it is essential to find an improvement goal that:

- It can be used to gain early wins.
- It is small enough to be achievable in a reasonable timeframe.
- Has benefits that are significant enough to be evident to the organization.

It allows constant learning from rapid feedback and recovering from mistakes quickly.

Note

The aim is to build a snowball effect where each new successful outcome adds to previous successful results. It will maximize the buy-in from all affected.

Continue

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

Identify project metrics and key performance indicators (KPIs)

Completed

- 3 minutes

We spoke earlier about the importance of shared goals. It was also agreed upon by team members that the goals needed to be specific, measurable, and time-bound.

It is essential to establish (and agree upon) appropriate metrics and Key Performance Indicators (KPIs) to ensure these goals are measurable.

While there is no specific list of metrics and KPIs that apply to all DevOps Projects, the following are commonly used:

Faster outcomes

- **Deployment Frequency** . Increasing the frequency of deployments is often a critical driver in DevOps Projects.
- **Deployment Speed** . It is necessary to reduce the time that they take.
- **Deployment Size** . How many features, stories, and bug fixes are being deployed each time?
- **Lead Time** . How long does it take from the creation of a work item until it is completed?

Efficiency

- **Server to Admin Ratio** . Are the projects reducing the number of administrators required for a given number of servers?
- **Staff Member to Customers Ratio** . Is it possible for fewer staff members to serve a given number of customers?
- **Application Usage** . How busy is the application?
- **Application Performance** . Is the application performance improving or dropping? (Based upon application metrics)?

Quality and security

- **Deployment failure rates** . How often do deployments (or applications) fail?
- **Application failure rates** . How often do application failures occur, such as configuration failures, performance timeouts, and so on?
- **Mean time to recover** . How quickly can you recover from a failure?
- **Bug report rates** . You do not want customers finding bugs in your code. Is the amount they are seeing increasing or lowering?
- **Test pass rates** . How well is your automated testing working?
- **Defect escape rate** . What percentage of defects are being found in production?
- **Availability** . What percentage of time is the application truly available for customers?
- **Service level agreement achievement** . Are you meeting your service level agreements (SLAs)?
- **Mean time to detection** . If there is a failure, how long does it take for it to be detected?

Culture

- **Employee morale** . Are employees happy with the transformation and where the organization is heading? Are they still willing to respond to further changes? This metric can be challenging to measure but is often done by periodic, anonymous employee surveys.
- **Retention rates** . Is the organization losing staff?

Note

It is crucial to choose metrics that focus on specific business outcomes and achieve a return on investment and increased business value.

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.

In which of the following choices would you find large amounts of technical debt?

○

Greenfield project.

○

Brownfield project.

○

Bluefield project.

2.

Which of the following choices would a system that manages inventory in a warehouse be considered?

○

System of Record.

○

System of Engagement.

○

System of History.

3.

Which of the following choices are the categorized user groups most adopted in Continuous Delivery?

○

Canaries, Early adopters, and Users.

○

Alpha and Beta Users.

○

Blue and Green Users.

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

In this module, you learned how to decide how to apply the DevOps process and tools to minimize initial resistance.

Also, how to identify teams, plan for a DevOps transformation culture, and define timelines for your 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

- Greenfield project - Wikipedia
- Brownfield (software development) - Wikipedia
- About teams & Agile tools | Microsoft Docs
- Best practices for Agile project management | Microsoft Docs

Continue