# Summary

* 5 minutes

In this module, you learned about what DevOps is (and what it's not) and about the Microsoft DevOps suite, Azure DevOps. You also set up an account in Azure DevOps and created an organization.

In a comparison between elite performers and low performers, elite performers deploy more frequently, more quickly, and with fewer failures. This mindset helps them better adapt to changing market conditions, experiment with new features, and recover from incidents with greater resiliency. DevOps gives you a path to becoming an elite performer.

Even for elite performers, change happens gradually, often starting with the most immediate challenges or pain points. Adopting DevOps practices takes time.

## Learn more

We'll go deeper into Azure DevOps in future modules. You can also check out these resources:

* What is DevOps?
* DevOps at Microsoft
* What features and services do I get with Azure DevOps?

## Appendix: Recommendations for KPIs and quality metrics

The following sections are optional. They provide additional information beyond what's covered in this module.

DevOps encourages teams to have goals that are specific, measurable, and time-bound. To ensure that these goals are measurable, it's important for the team to agree on appropriate metrics and Key Performance Indicators (KPIs). It's important to choose metrics that focus on specific business outcomes, and that achieve a return on investment and increased business value. Here is a list of commonly used metrics and KPIs that apply to all DevOps teams:

### Faster Outcomes

* Deployment Frequency. Increasing the frequency of deployments is often a critical driver in DevOps teams.
* Deployment Speed. As well as increasing how often deployments happen, it's important to decrease 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 starting on a work item, until it is deployed?

### 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 (and/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 don't want customers finding bugs in your code. Is the amount they are finding increasing or decreasing?
* 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?
* SLA 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?
* Retention Rates. Is the organization losing staff?

One of the promises of DevOps is to deliver software faster and with higher quality. It may seem that these promises are contradictory. Traditionally, the faster you delivered the software, the lower the quality. Higher-quality software always took longer. However, DevOps processes can help you to find problems earlier, and this usually means that they take less time to fix.

### Common quality metrics

* Failed Builds Percentage - Overall, what percentage of builds are failing?
* Failed Deployments Percentage - Overall, what percentage of deployments are failing?
* Ticket Volume - What is the overall volume of customer bug tickets?
* Bug Bounce Percentage - What percentage of customer or bug tickets are being reopened?
* Unplanned Work Percentage - What percentage of the overall work being performed is unplanned?