**What is Azure DevOps?**

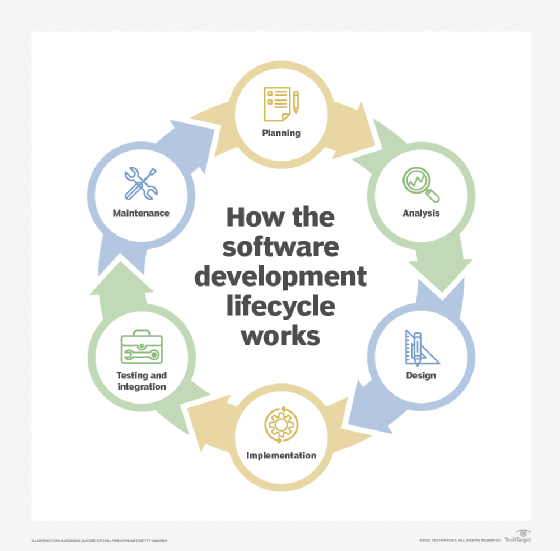
Azure DevOps -- rebranded from Visual Studio Team Services (VSTS) in 2018 -- is a software as a service ([SaaS](https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service)) platform from Microsoft designed to provide a comprehensive [toolchain](https://www.techtarget.com/searchsoftwarequality/definition/software-toolchain) for developing and deploying software projects.

The platform assists [development teams](https://www.techtarget.com/searchapparchitecture/tip/A-primer-on-core-development-team-structure-concepts) and offers special tools and services for software programmers, analysts and testers as well as IT project or team managers. Azure DevOps integrates with a wide range of other tools, [expanding DevOps toolsets](https://www.techtarget.com/searchitoperations/definition/DevOps) and tailoring them to the unique needs of the development team and organization.

Azure DevOps is an end-to-end software development platform that offers an assortment of capabilities intended to organize and accelerate development efforts across the entire application lifecycle:

* Requirements management.
* Project management for both [Agile software development and waterfall](https://www.theserverside.com/tip/Agile-vs-Waterfall-Whats-the-difference?) teams.
* Version control using Team Foundation Version Control (TFVC) or Git.
* Automated builds.
* Reporting, such as test results, and development metrics, such as backlogs and release velocity.
* Testing and release management.

Azure DevOps is intended for use with native [development environments](https://www.techtarget.com/searchsoftwarequality/definition/development-environment) such as Microsoft Visual Studio and [Eclipse](https://www.techtarget.com/searchapparchitecture/definition/Eclipse-Eclipse-Foundation). However it can be used as a backend to other popular integrated development environments ([IDEs](https://www.techtarget.com/searchsoftwarequality/definition/integrated-development-environment)).

The Azure

DevOps software development platform offers tools to aid development efforts across the entire application lifecycle.

**What services does Azure DevOps offer?**

Azure DevOps offers a suite of focused services, with each service related to a key portion of the development lifecycle.

**Azure Boards for project management.** This scalable service tracks the team's work with Agile and [Scrum](https://www.techtarget.com/searchsoftwarequality/definition/Scrum) techniques or [Kanban](https://www.techtarget.com/whatis/definition/kanban) boards, project backlogs, customized reporting and issue tracking, and team dashboards. For example, developers can use Azure Boards to do the following:

* Follow user stories, bugs and features.
* Access interactive backlogs, boards, lists and calendar views.
* Create item worklists and charts.
* Develop delivery plans that consider dependencies.
* Connect with [GitHub](https://www.techtarget.com/searchitoperations/definition/GitHub) repositories to link GitHub commits, pull requests and issues.

Azure DevOps provides end-to-end traceability, which lets developers track work throughout the entire lifecycle from requirements through deployment.

**Key Components of Azure DevOps**

Azure DevOps is composed of several services, each catering to a specific aspect of the software development lifecycle:

* **Azure Repos**: A set of version control tools that provide both Git (distributed) and Team Foundation Version Control (TFVC) (centralized) options. Developers can collaborate, track changes, and manage versions of their code effectively.
* **Azure Pipelines**: A CI/CD service that automates the building, testing, and deployment of applications. It supports multiple languages and frameworks and integrates with various platforms like GitHub, Docker, and Kubernetes.
* **Azure Boards**: A work tracking system that includes Kanban boards, backlogs, team dashboards, and custom reporting. Azure Boards help teams plan, track, and discuss work across the entire development process.
* **Azure Test Plans**: A testing solution that provides a comprehensive suite of testing tools for manual, exploratory, and automated testing. It enables continuous testing with real-time insights.
* **Azure Artifacts**: A package management service that allows teams to create, host, and share packages (e.g., NuGet, npm, Maven, etc.) across the organization. It integrates seamlessly with Azure Pipelines for package versioning and distribution.
* **Azure DevTest Labs**: A service that allows developers to quickly create environments in Azure to develop and test applications. These environments can be automated, configured with reusable templates, and managed efficiently to optimize costs.

**Benefits of Azure DevOps**

* **End-to-End DevOps Solution**: Azure DevOps covers the entire software development lifecycle, providing a unified platform for development teams to collaborate and deliver software faster.
* **Scalability and Flexibility**: Azure DevOps can scale according to the needs of the organization, whether it's a small team or a large enterprise. It supports various development environments and integrates with many popular tools.
* **Cloud-Agnostic and Extensible**: While it’s a part of Microsoft Azure, Azure DevOps can be used with any cloud provider or on-premises environment. It is also extensible, allowing teams to integrate their preferred tools and services via a marketplace that offers numerous extensions.
* **Security and Compliance**: Azure DevOps includes features to ensure security throughout the development process, including permissions management, secret management, and compliance with industry standards.

**Use Cases for Azure DevOps**

* **CI/CD Automation**: Automating the build, test, and deployment process across different environments.
* **Agile Project Management**: Managing projects using agile methodologies, tracking work items, sprints, and backlogs.
* **Collaborative Development**: Enabling teams to work together on code, track changes, and manage branches and pull requests.
* **Infrastructure as Code**: Managing and deploying infrastructure using code, often integrated with tools like Terraform or ARM templates.