Brückenbauer zwischen Technologie, Wirtschaft & Politik

David Gierse

Berlin, der 31.01.2025

Optimale Nutzung von Azure DevOps: Das müsst ihr wissen

Einführung in Azure DevOps

Azure DevOps ist eine Plattform von Microsoft, die eine vollständige Suite von Entwicklungswerkzeugen bereitstellt. Sie unterstützt Teams bei der Planung, Entwicklung, Bereitstellung und Wartung von Software. Durch die enge Integration verschiedener Services bietet Azure DevOps eine durchgängige Lösung für moderne Softwareentwicklungsprozesse.

Tipp: Starte mit kleinen Projekten, um dich mit den verschiedenen Tools und Funktionen vertraut zu machen, bevor du größere Vorhaben angehst.

Zielsetzung von Azure DevOps

Das Hauptziel von Azure DevOps ist die **Optimierung des gesamten Softwareentwicklungsprozesses** durch eine nahtlose Integration von Entwicklungs- und Betriebsteams. Es fördert **kollaboratives Arbeiten** und reduziert Fehler durch Automatisierung und Strukturierung.

Tipp: Definiere klare Ziele und Erwartungen für dein Team, um sicherzustellen, dass alle Beteiligten auf ein gemeinsames Ziel hinarbeiten. Eine gut definierte Roadmap kann Missverständnisse vermeiden und Produktivität steigern.

Azure Boards

Azure Boards bieten ein flexibles und leistungsstarkes **Kanban- und Scrum-Board** zur Verwaltung von Work Items, Epics, Features und User Stories. Es ermöglicht eine detaillierte Nachverfolgung von Aufgaben und deren Fortschritt innerhalb eines Projekts.

Vorteile:

- Effiziente Verwaltung von Work Items
- Bessere Visualisierung von Aufgaben

https://www.linkedin.com/in/david-gierse-49a34718b/

- Klare Strukturierung von Projekten

Tipp: Verwende Tags und benutzerdefinierte Felder, um Work Items besser zu organisieren und den Überblick zu behalten. Achte darauf, regelmäßig Statusupdates zu geben, um die Transparenz im Team zu verbessern.

Do's: ✅ Nutze die Board-Funktion zur besseren Aufgabenverteilung und Priorisierung. ✅ Implementiere automatisierte Workflows zur schnelleren Bearbeitung.

Don'ts: ❌ Vermeide es, Work Items ohne klare Verantwortlichkeiten zu erstellen. ❌ Verzichte nicht auf regelmäßige Überprüfung und Anpassung der Aufgabenlisten.

Pipelines

Azure Pipelines ermöglichen eine **automatisierte Continuous Integration (CI) und Continuous Deployment (CD)**. Dadurch wird sichergestellt, dass Software-Updates effizient und fehlerfrei in die Produktion überführt werden.

Vorteile:

- Unterstützung für Drittanbieter-Plugins
- Nahtlose Integration von Anwendungen
- Regelmäßige Automatisierung zur Vermeidung von Fehlern und zur Verbesserung der Codequalität

Tipp: Implementiere regelmäßige automatisierte Tests in deinen Pipelines, um frühzeitig Fehler zu erkennen. Konfiguriere verschiedene Umgebungen (Dev, Test, Prod), um den Deployment-Prozess sicherer zu gestalten.

Do's: ✅ Nutze automatisierte Tests zur Codeprüfung. ✅ Implementiere ein Rollback-System, um fehlerhafte Deployments rückgängig zu machen.

Don'ts: ❌ Vermeide manuelle Deployments, da sie fehleranfällig sind. ❌ Ignoriere keine Fehlermeldungen – überprüfe Logs und behebe Probleme frühzeitig.

Epics, Features und User Stories

- **Epics:** Große Funktionsbereiche oder Projekte.
- **Features:** Komponenten innerhalb eines Epics.
- **User Stories:** Funktionen aus Sicht des Endbenutzers.

Brückenbauer zwischen Technologie, Wirtschaft & Politik

Tipp: Verwende die **SMART-Kriterien**, um klare und umsetzbare Ziele zu formulieren. Achte darauf, dass User Stories einen echten Mehrwert liefern und nicht zu vage formuliert sind.

Do's: ✅ Erstelle gut definierte User Stories mit klaren Anforderungen. ✅ Priorisiere Aufgaben basierend auf Business-Value und Nutzerfeedback.

Don'ts: ❌ Vermeide unklare Epics ohne eindeutige Zielsetzung. ❌ Erstelle keine zu umfangreichen User Stories – teile große Aufgaben in kleinere Arbeitsschritte.

Repositories

Azure Repositories ermöglichen eine **strukturierte Codeverwaltung** mit Branching-Strategien und Pull Requests.

Tipp: Nutze **Feature Branches und Pull Requests**, um den Entwicklungsprozess zu optimieren und die Codequalität zu sichern.

Do's: ✅ Implementiere Code Reviews als festen Bestandteil des Entwicklungsprozesses. ✅ Nutze eine klare Naming-Konvention für Branches, um Übersichtlichkeit zu gewährleisten.

Don'ts: ❌ Vermeide unstrukturierte Commits ohne klare Nachrichten. ❌ Verzichte nicht auf regelmäßige Backups deines Codes.

Artifacts

Azure Artifacts ermöglichen das **Erstellen, Hosten und Teilen von Paketen** innerhalb von Entwicklungsprojekten. Es erleichtert das Dependency Management und fördert die Wiederverwendung von Code.

Tipp: Setze die richtigen **Feed-Berechtigungen**, um den Zugriff sicher zu steuern.

Do's: ✅ Nutze Artifacts, um Abhängigkeiten sauber zu verwalten. ✅ Dokumentiere die verwendeten Pakete für eine bessere Nachverfolgbarkeit.

Don'ts: ❌ Vermeide nicht autorisierte Änderungen an gemeinsam genutzten Paketen. ❌ Ignoriere keine Sicherheitsupdates für externe Abhängigkeiten.

Testpläne

Azure Test Plans bietet eine **umfassende Testmanagement-Lösung**, um manuelle und automatisierte Tests zu planen und durchzuführen.

https://www.linkedin.com/in/david-gierse-49a34718b/

Brückenbauer zwischen Technologie, Wirtschaft & Politik

Tipp: Kombiniere **manuelle und automatisierte Tests**, um eine bessere Testabdeckung zu erreichen.

Do's: ✅ Definiere Testfälle klar und verständlich. ✅ Automatisiere wiederholbare Tests zur Effizienzsteigerung.

Don'ts: ❌ Vermeide unzureichende Testabdeckung. ❌ Verzichte nicht auf regelmäßige Tests nach Änderungen im Code.

Tipps und Tricks

Best Practices und Empfehlungen:

- **Regelmäßige Reviews** durchführen, um den Fortschritt zu bewerten.
- **Automatisierte Tests** für mehr Effizienz.
- **Prozesse klar strukturieren** und standardisieren.
- **Monitoring-Tools und Dashboards** für eine bessere Visualisierung wichtiger Kennzahlen einsetzen.

Fazit

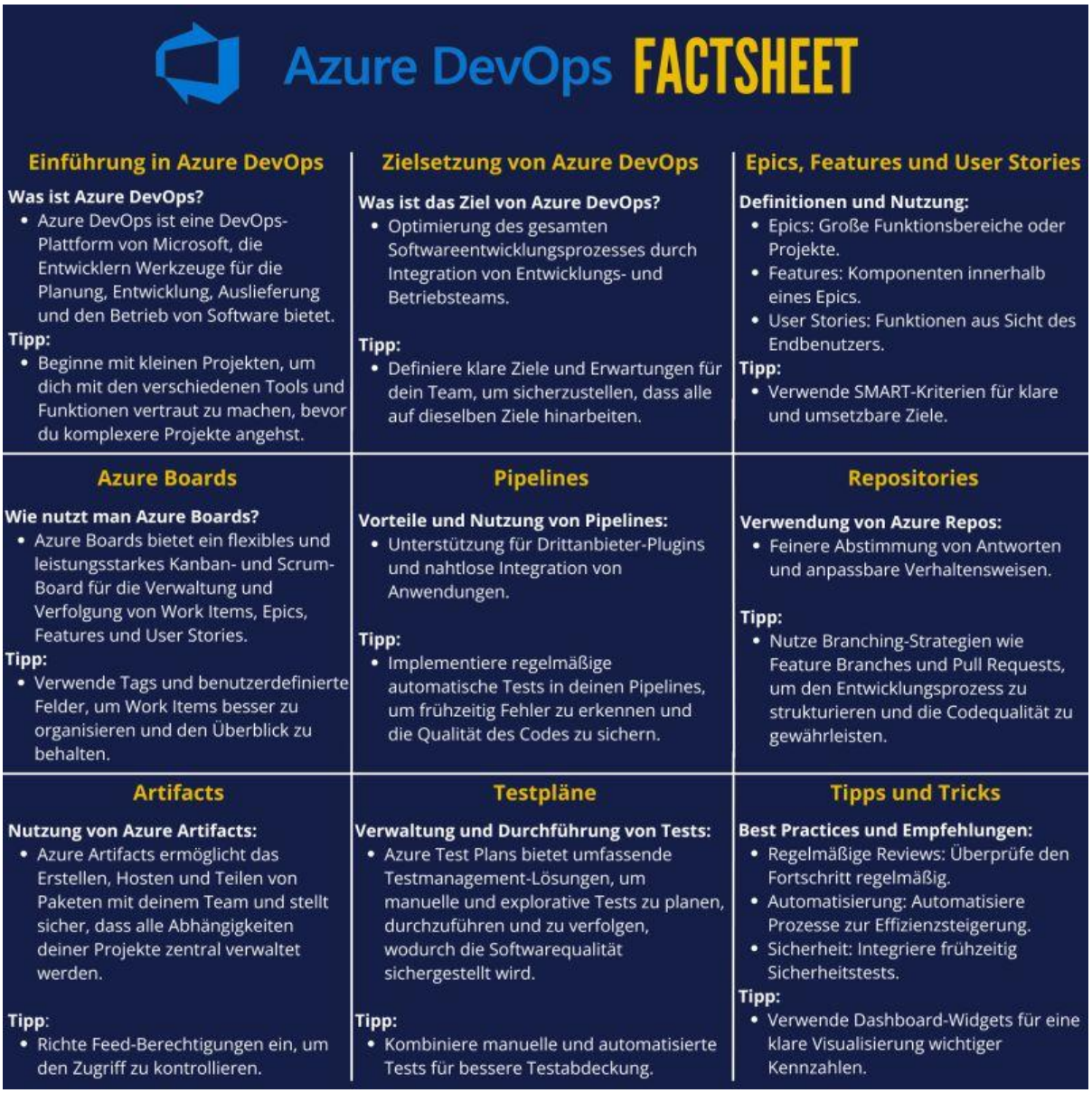
Azure DevOps bietet eine umfassende Lösung für die **moderne Softwareentwicklung**. Durch die Nutzung von Azure Boards, Pipelines, Repos und Testplänen können Teams ihre Effizienz steigern und ihre Projekte besser verwalten.

Mit einer gut durchdachten Nutzung der verschiedenen Funktionen und der Umsetzung bewährter Praktiken können Entwicklerteams die Vorteile von Azure DevOps voll ausschöpfen.

Welche Erfahrungen habt ihr mit Azure DevOps gemacht? Welche Best Practices könnt ihr empfehlen? Schreibt eure Meinungen in die Kommentare!

https://www.linkedin.com/posts/david-gierse-49a34718b_entwicklung-artifacts-azuredevops-activity-7201505458610667521-kAvk?utm_source=share&utm_medium=member_desktop

https://www.linkedin.com/in/david-gierse-49a34718b/


Azure DevOps FACTSHEET
Einführung in Azure DevOps
Was ist Azure DevOps?
• Azure DevOps ist eine DevOps-Plattform von Microsoft, die Entwicklern Werkzeuge für die Planung, Entwicklung, Auslieferung und den Betrieb von Software bietet.
Tipp:
• Beginne mit kleinen Projekten, um dich mit den verschiedenen Tools und Funktionen vertraut zu machen, bevor du komplexere Projekte angehst.
Zielsetzung von Azure DevOps
Was ist das Ziel von Azure DevOps?
• Optimierung des gesamten Softwareentwicklungsprozesses durch Integration von Entwicklungs- und Betriebsteams.
Tipp:
• Definiere klare Ziele und Erwartungen für dein Team, um sicherzustellen, dass alle auf dieselben Ziele hinarbeiten.
Epics, Features und User Stories
Definitionen und Nutzung:
• Epics: Große Funktionsbereiche oder Projekte.
• Features: Komponenten innerhalb eines Epics.
• User Stories: Funktionen aus Sicht des Endbenutzers.
Tipp:
• Verwende SMART-Kriterien für klare und umsetzbare Ziele.
Azure Boards
Wie nutzt man Azure Boards?
• Azure Boards bietet ein flexibles und leistungsstarkes Kanban- und Scrum-Board für die Verwaltung und Verfolgung von Work Items, Epics, Features und User Stories.
Tipp:
• Verwende Tags und benutzerdefinierte Felder, um Work Items besser zu organisieren und den Überblick zu behalten.
Pipelines
Vorteile und Nutzung von Pipelines:
• Unterstützung für Drittanbieter-Plugins und nahtlose Integration von Anwendungen.
Tipp:
• Implementiere regelmäßige automatische Tests in deinen Pipelines, um frühzeitig Fehler zu erkennen und die Qualität des Codes zu sichern.
Repositories
Verwendung von Azure Repos:
• Feinere Abstimmung von Antworten und anpassbare Verhaltensweisen.
Tipp:
• Nutze Branching-Strategien wie Feature Branches und Pull Requests, um den Entwicklungsprozess zu strukturieren und die Codequalität zu gewährleisten.
Artifacts
Nutzung von Azure Artifacts:
• Azure Artifacts ermöglicht das Erstellen, Hosten und Teilen von Paketen mit deinem Team und stellt sicher, dass alle Abhängigkeiten deiner Projekte zentral verwaltet werden.
Tipp:
• Richte Feed-Berechtigungen ein, um den Zugriff zu kontrollieren.
Testpläne
Verwaltung und Durchführung von Tests:
• Azure Test Plans bietet umfassende Testmanagement-Lösungen, um manuelle und explorative Tests zu planen, durchzuführen und zu verfolgen, wodurch die Softwarequalität sichergestellt wird.
Tipp:
• Kombiniere manuelle und automatisierte Tests für bessere Testabdeckung.
Tipps und Tricks
Best Practices und Empfehlungen:
• Regelmäßige Reviews: Überprüfe den Fortschritt regelmäßig.
• Automatisierung: Automatisiere Prozesse zur Effizienzsteigerung.
• Sicherheit: Integriere frühzeitig Sicherheitstests.
Tipp:
• Verwende Dashboard-Widgets für eine klare Visualisierung wichtiger Kennzahlen.