

What is DevOps and Azure DevOps

And what is not DevOps

Subodh Sohoni



Application Lifecycle Management



How DevOps has become essential



Definition of DevOps

DevOps is a software **development and delivery process** that emphasizes

- **communication and collaboration** between product management, software development, operations professionals
- and close alignment with **business objectives**.

It supports this by **automating** and monitoring the process of software integration, testing, deployment, and infrastructure changes

- by establishing a **culture** and environment
- where building, testing, and releasing software can happen **rapidly, frequently, and more reliably**

Mindsets

Developers

- Implement every change demanded by customer
- Make changes rapidly

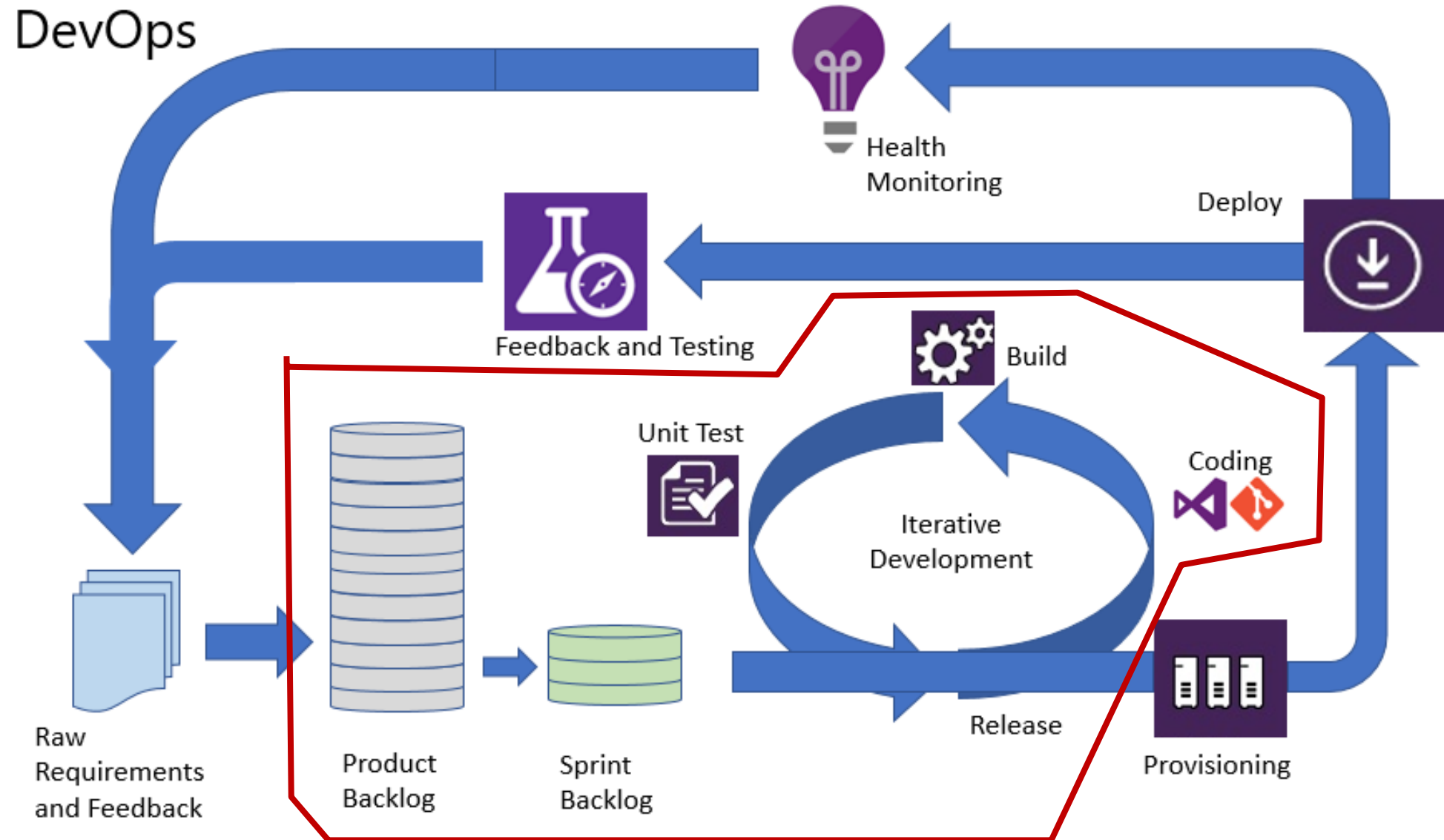
Operations

- Do not change status quo if nothing is broken
- Change has potential to break

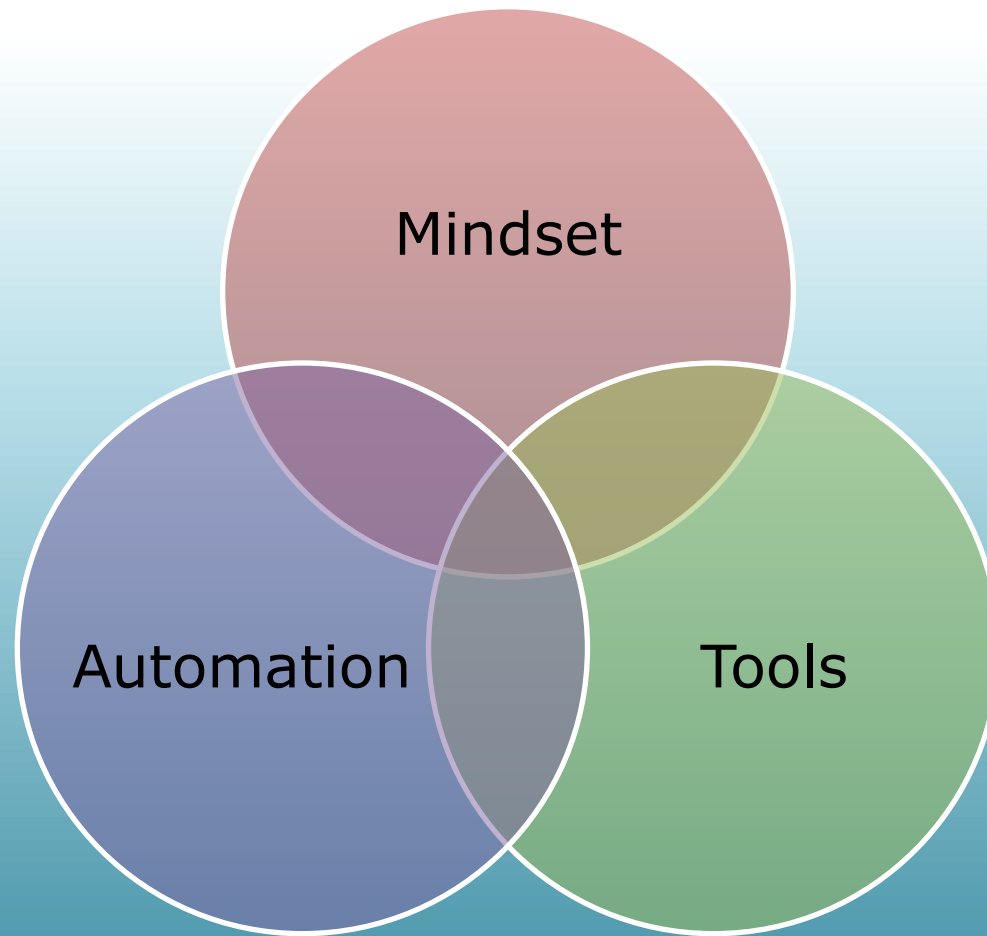
Testers

- Responsibility of quality is only mine
- I don't give importance to delivery over quality

My interpretation of what DevOps means



Pillars of DevOps

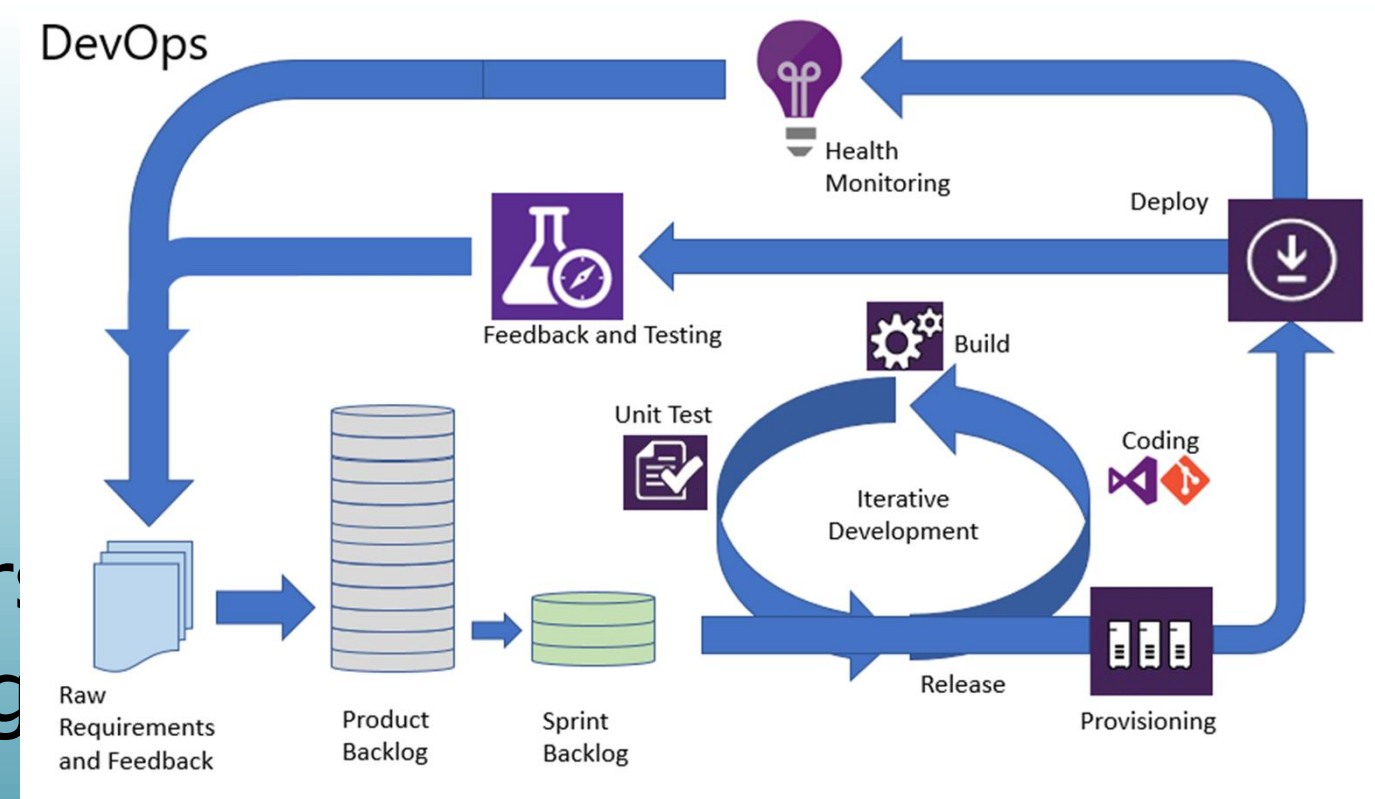


What is not (only) DevOps

- Is DevOps engineer job description or designation?
- DevOps is not an invasion of developers over operations
- DevOps is not only Automation and not only Culture and not just the Tools

Which roles combine to form a DevOps team

- Business
- Architects
- Developers
- Testers
- DBA
- System Administrator
- Build and Release Eng
- Management



Goals to achieve with DevOps

Improve deployment frequency

Achieve faster time to market

Lower failure rate of new releases

Shorten lead time between fixes

Improve mean time to recovery

DevOps Practices

Self-service
configuration

Automated
provisioning

Automated
release
management

Continuous
integration

Continuous
deployment

Continuous
testing

Azure DevOps Tools

Boards - Agile Planning and Monitoring

Repo - Version Control

Pipelines - Process automation with Build, test, and deployment.

Automated platform provisioning

Code promotion and synchronization across environments and servers

Dependency analysis and impact analysis

Application Insights - Health Monitoring

Pitfalls to be avoided in DevOps implementation

- Focus only on one of the aspects of DevOps – Tools, Automation or Culture
- Make operations team common to all development teams by Creating a separate special “DevOps Team”
- Choosing speed over quality
- Ignoring Database – DataOps
- Forgetting Security
- Not involving all concerned persons and stakeholders early on
- Not understanding that Scrum ≠ Agile ≠ DevOps
- Failing to look at the larger picture of ALM

Dashboard and Wiki

- Dashboards - Information emitters – Show widgets of your choice
- Wiki – Internal discussion board for the team.

Packaging – Artifacts Service

- Create and publish Packages that can be used by other teams in the organization
- Supported Technologies:
 - NuGet
 - NPM
 - Maven
 - Python
- Create your own feed for the organization

Test Hub - Test Planning Service

Test Plan – Plan which tests to execute per sprint

Test Suite – Make a group of test cases

Test Case – Steps and Expected results. Parameters for showing which data to enter

Shared Steps

Test Runner – Run the test case manually

Create Bug – One step process to file a bug while testing. Automatically enters required data like steps to reproduce, screen snapshots, screen video etc.