



DevOps

► Introduction

Nagarathna

Agenda

- ▶ What is DevOps?
- ▶ Why DevOps?
- ▶ Life Cycle
- ▶ Agile Vs DevOps
- ▶ DevOps - Metrics and Tools
- ▶ DevOps Values
- ▶ Fast Delivery Cycles
- ▶ Practices
- ▶ Why DevOps is popular
- ▶ Habits of DevOps
- ▶ Build and release pipeline
- ▶ Pipeline steps
- ▶ Types of testing

DevOps

- ▶ *"DevOps is the union of people, process, and products to enable continuous delivery of value to our end users."*

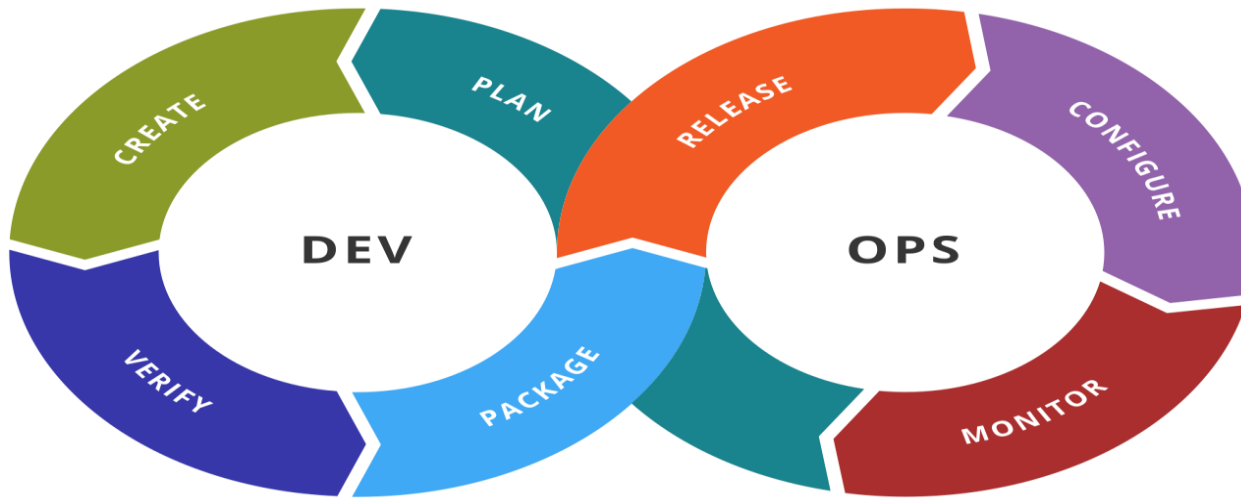
DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.



Why is DevOps used?

- ▶ **Predictability:** Offers significantly lower failure rate of new releases
- ▶ **Reproducibility:** Version everything so that earlier version can be restored anytime.
- ▶ **Maintainability:** Effortless process of recovery in the event of a new release crashing or disabling the current system.
- ▶ **Time to market:** Reduces the time to market
- ▶ **Greater Quality:** Improve quality of application development as it incorporates infrastructure issues.
- ▶ **Reduced Risk:** It helps in reduction of defects across the lifecycle.
- ▶ **Resiliency:** The Operational state of the software system is more stable, secure, and changes are auditable.
- ▶ **Cost Efficiency**
- ▶ **Breaks larger code base into small pieces:** DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks.

DevOps Lifecycle



DevOps

- ▶ Software development method which focuses on **communication, integration, and collaboration (b/w dev & operations)**.
- ▶ **Enables** rapid **deployment** of products or services.
- ▶ **Culture**
- ▶ Deploying to production in a faster and automated way.

Agile

- ▶ Involves continuous iteration of development and testing in the SDLC process.
- ▶ Iterative, incremental, and evolutionary development.
- ▶ Breaks the product into smaller pieces and integrates them for final testing.

Agile Vs. DevOps



Cont..



Agile development involves a set of practices such as: *Agile Scrum & Agile Kanban*.

Cont..



DevOps involves a set of technical processes such as: Continuous Development, Continuous Integration (CI), Continuous Testing (CI), Continuous Deployment (CD) and Continuous Monitoring.

Difference Between Agile and DevOps

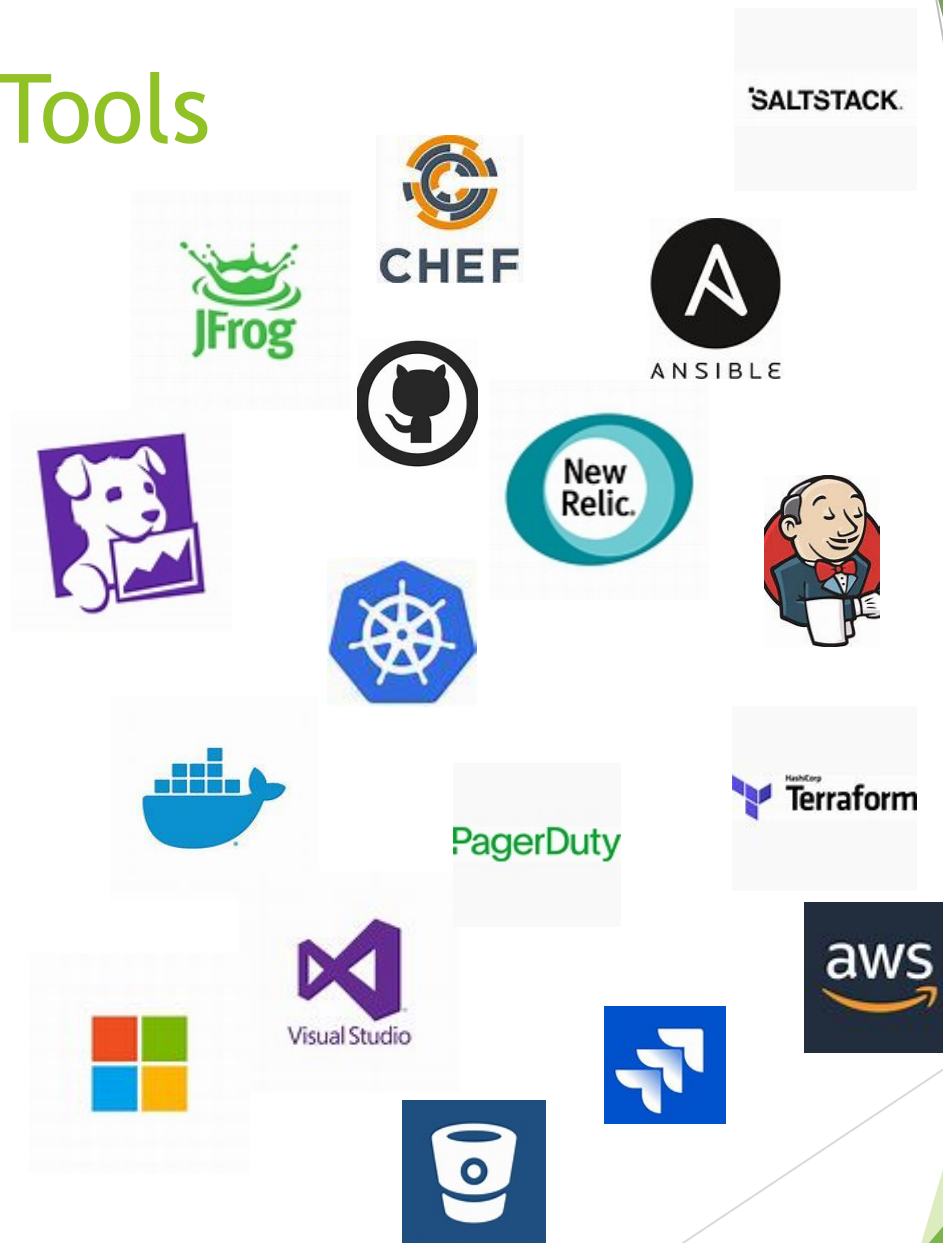
- ▶ Purpose is - Agile helps to manage complex projects.
- ▶ Target area - is Software Development.
- ▶ offers shorter development cycle and improved defect detection.
- ▶ Developing software is inherent to Agile
- ▶ It focuses on functional and non-function readiness.
- ▶ Agile doesn't emphasize on automation. Though it helps.
- ▶ In Agile, feedback is mostly given by customers.
- ▶ Central concept is to manage end-to-end engineering process
- ▶ End-to-end business solution and fast delivery.
- ▶ Supports Agile's release cycle.
- ▶ Developing, testing and implementation all are equally important.
- ▶ It focuses more on operational and business readiness.
- ▶ Automation is the primary goal of DevOps. It works on the principle to maximize efficiency when deploying software.
- ▶ Feedback is mostly measured by the internal team (by using Continuous Monitoring tools).

DevOps - Metrics

- ▶ *Deployment (or Change) Frequency*
- ▶ *Change Lead Time*
- ▶ *Change Failure Rate*
- ▶ *Mean Time To Recover (MTTR)*

DevOps - Tools

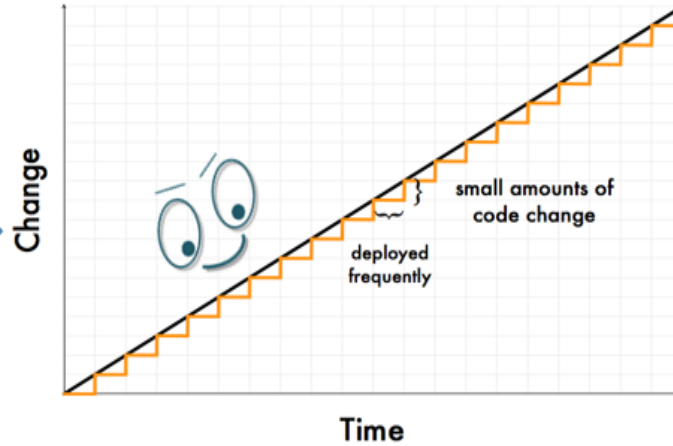
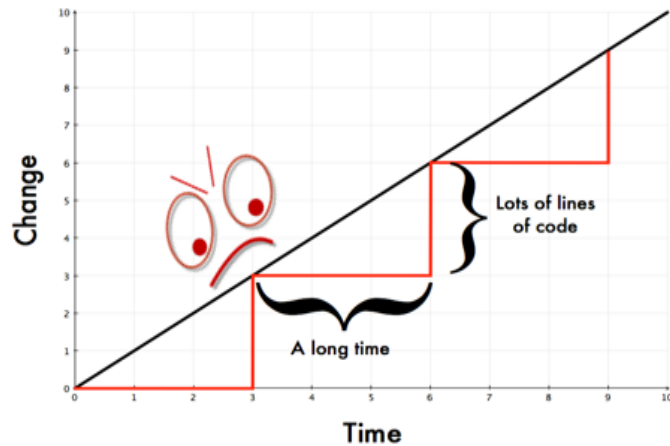
- ▶ GIT
- ▶ JENKINS
- ▶ SELENIUM
- ▶ DOCKER
- ▶ PUPPET
- ▶ CHEF
- ▶ ANSIBLE
- ▶ NAGIOS
- ▶ ELK STACK
- ▶ SPLUNK



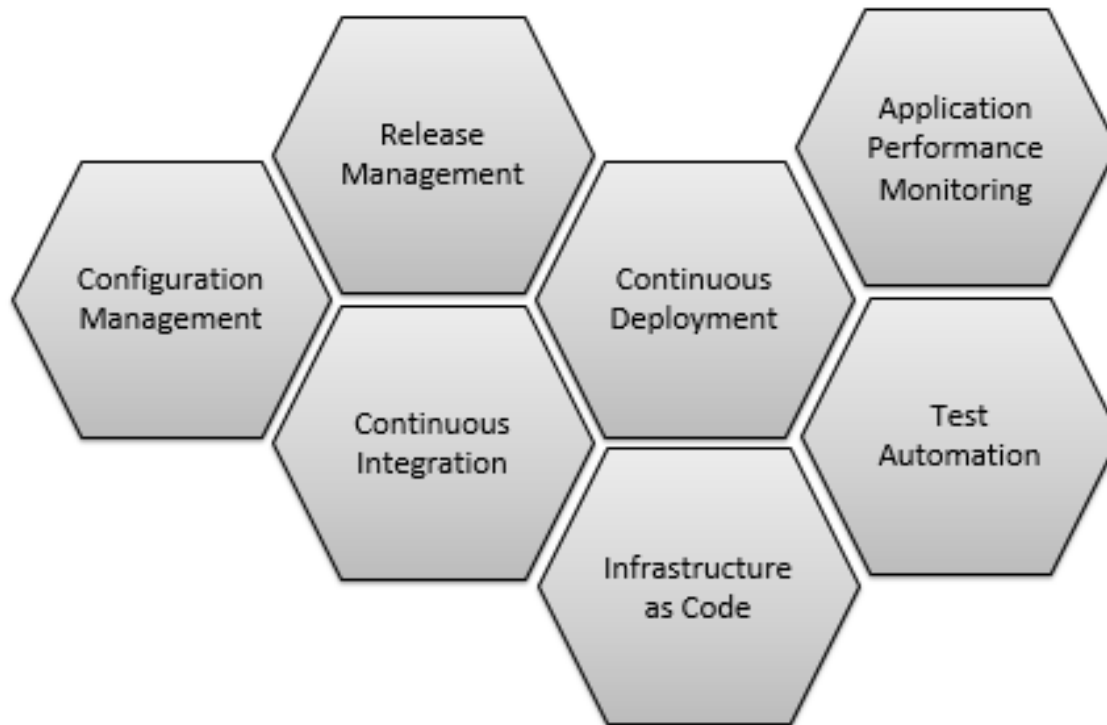
DevOps - Values

- ▶ How do we improve product delivery to our customers?
- ▶ How do we change product more quickly to better satisfy our customers?
- ▶ How do we recover after failing our customers?
- ▶ How do we get paid faster by our customers?

Fast Delivery Cycles



Practices

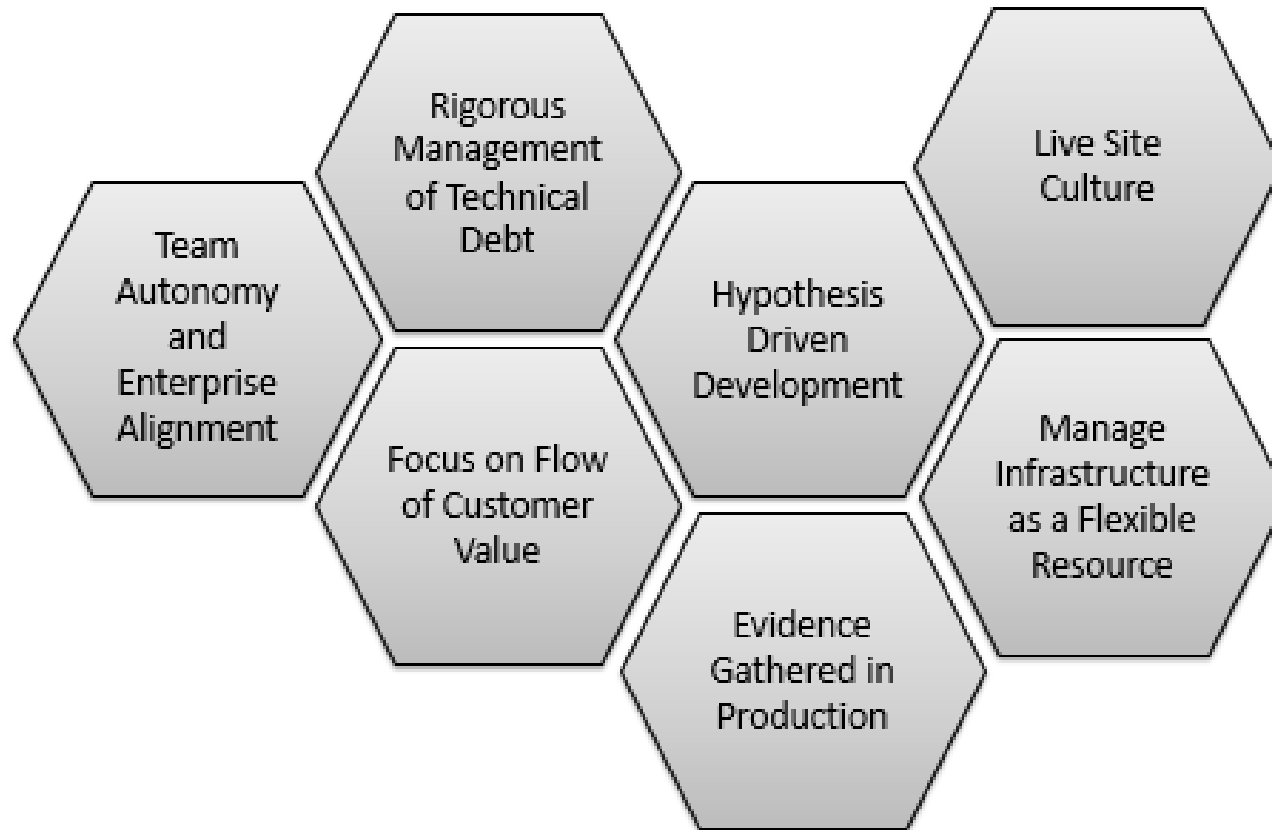


Why Devops is Popular

- ▶ Shorter Development Cycles
- ▶ Reduce Implementation Failure
- ▶ Better Communication and Cooperation

Habits of successful Devops

Habits



Habits of Devops

► Team Autonomy and Enterprise Alignment

- Common backlog & Self driven Scrum Teams
- 4 x 4 Planning

► Rigorous Management of Technical Debt

- Continuous monitoring of development
- Identifying tech debt - Bug, architecture issues
- Minimum no. of bugs

► Focus on the Flow of Customer Value

- Practice of updating & maintaining production server
- observe what gets used, what doesn't get used
- Looking at high usage account

Habits of Devops

▶ Hypothesis Driven Development

- Experiments to validate or disprove a hypothesis
- Delivers a measurable conclusion and enables continued learning.
- Enables continuous feedback from the key stakeholder

▶ Evidence Gathered in Production

- Performance measurement
- What went right
- What went wrong

▶ Live Site Culture

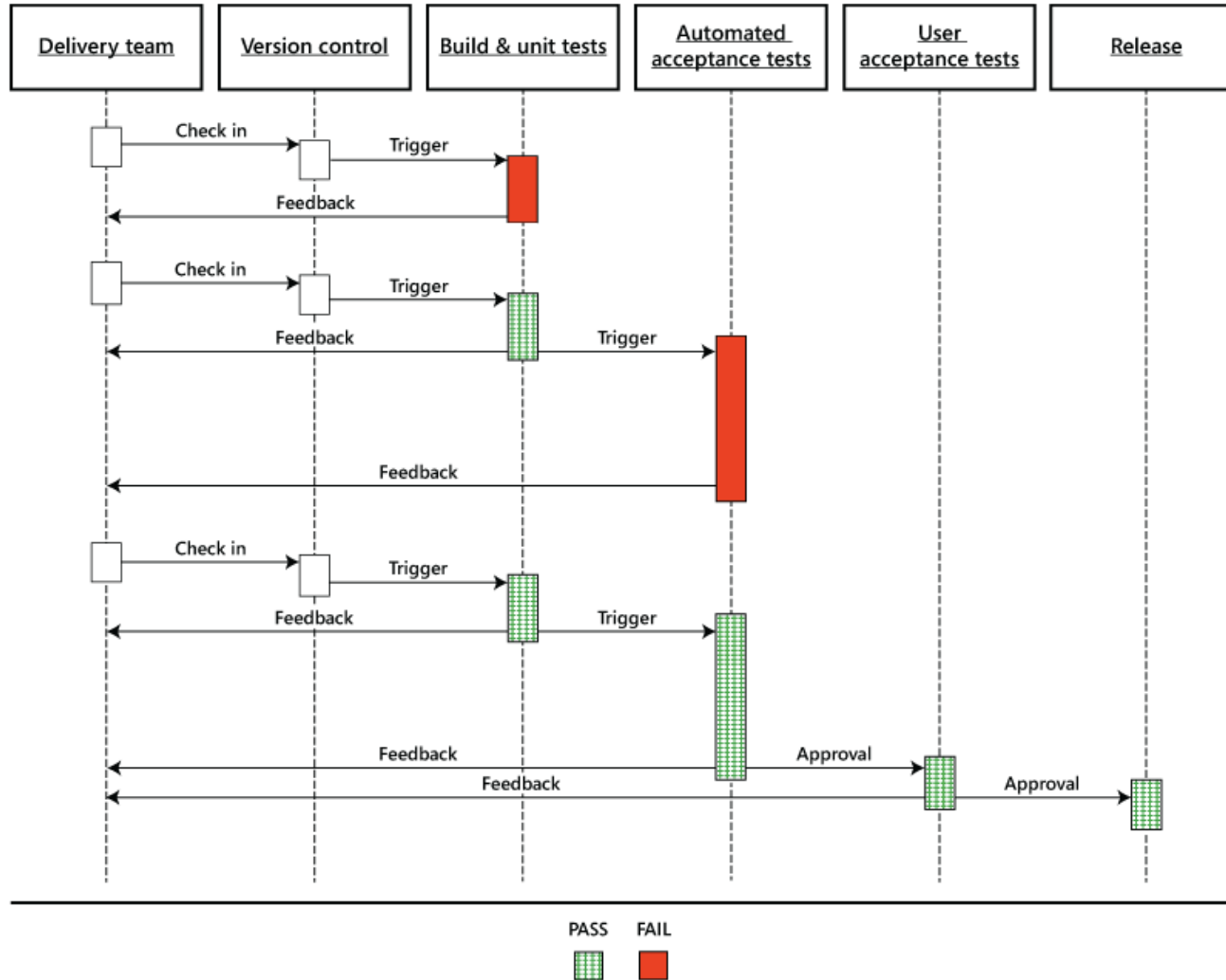
- Quick fix
- Root cause analysis

▶ Infrastructure as a Flexible Resource

IAC - manage your IT infrastructure using configuration files

Build and Release Pipeline

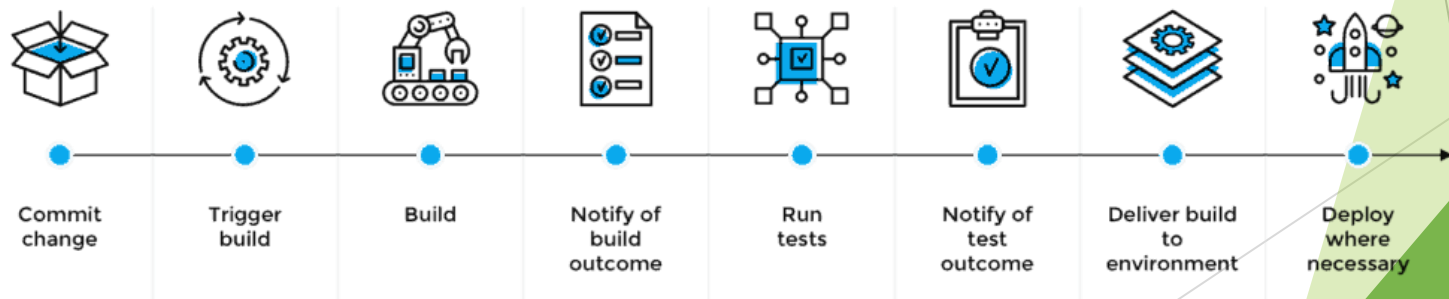
Build and Release Pipeline



What is a pipeline?

- ▶ A key practice in DevOps is continuous integration and continuous deployment (CI/CD).
- ▶ Is where the environments have different stages (e.g., dev, code analysis, testing, UAT, production), manual things are automated, and developers can achieve high-quality code, flexibility, and numerous deployments.
- ▶ Continuous integration - Build Pipeline
- ▶ Continuous Delivery/Deployment - Release Pipeline

CI/CD Pipeline



Why create a pipeline?

- ▶ A Pipeline's code defines your entire build process, which typically includes stages for building an application, testing it and then delivering it.
- ▶ DevOps says - maximum automation in a Software Development Life Cycle.
- ▶ Pipelines help in achieving this and keeping track of software releases.

Steps involved: Build Pipeline

1. Version Control:

- ▶ When a commit to the source branch is made, first stage of the build pipeline is triggered.

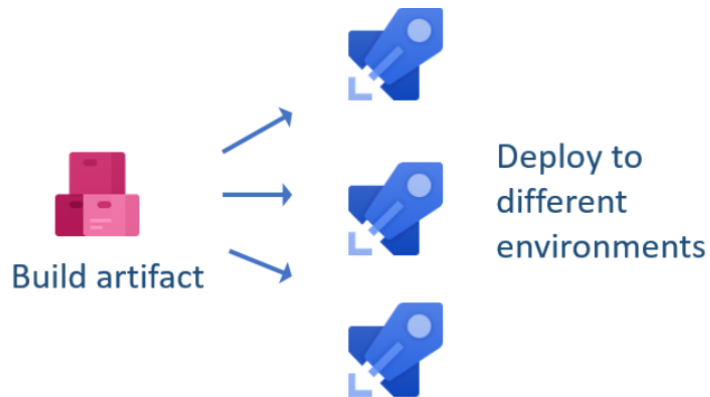
2. Unit and Integration Tests:

- ▶ Run unit tests on individual features developed.
- ▶ Run integration tests one more than one feature after integration.

Steps involved: Build Pipeline

3. Build :

- ▶ Build code.
- ▶ If all these steps are executed successfully, the executables are stored into an artifact repository for later use.



Steps involved: Release Pipeline

4. Automated Acceptance Tests:

- ▶ Acceptance testing is a process of running a series of tests over compiled/built code to test against the predefined acceptance criteria set by the business.

5. User Acceptance Test:

- ▶ During UAT, users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications.
- ▶ If all steps are executed successfully, deploy to customers.

Types of Testing

Major Classification

► Functional testing:

- Feeding a specific input and examining output
- Is conducted to evaluate the compliance of a system or component with specified functional requirements.
- Testers do not necessarily need to know how the software has been implemented

► Non-Functional testing:

- Tests software for the way it operates
- Reliability

Functional Testing

► Smoke Test:

- Tests the most important functionality of a component or system
- Identify main issues of the functionality

► Sanity Test:

- The point of a sanity test is to rule out certain obvious false results, not to catch every possible error.

► Regression Test:

- Extensive, repeatable, and automated testing of the entire software.

Non-Functional Testing

- ▶ Load Testing:

- ▶ Process of putting demand on a system and measuring its response.

- ▶ Performance Testing:

- ▶ How a system performs in terms of responsiveness and stability under a workload.

- ▶ Security testing:

- ▶ Determines that the data and resources of the system are protected from possible intruders.

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

Thank you