UNIT- II

✪ Software Development Models & DevOps :

Top - 8 Software Development Models Include,

→ Waterfall Model

→ V- Model

→ Incremental Model

→ RAD Model

→ Iterative Model

→ Spiral Model

→ Prototype Model

→ Agile Modile.

Is DevOps a s/w Development Model ?

DevOps is a Set of Practices , Tools , & a cultural philosophy that automate & integrate the processes b/w s/w development & IT Teams.

It Emphasizes Team Empowerment , Cross Team Communication & Collaboration & Technology Automation.

* What is the SDLC Model in DevOps?

The Software Development Lifecycle (SDLC) is the Cost - effective & Time - efficient process that development teams use to design & build high quality s/w.

The Goal of SDLC is to minimize Project risks through forward planning so that s/w meets customer expectations during production & beyond.

⊛ DevOps Lifecycle for Business Agility:

What are the 7C's of DevOps life cycle for Business Agility?

The 7C's of the DevOps lifecycle are,

→ Continuous Development (Coding - Git Hub)

→ Continuous Integration (Test & update code - Selenium)

→ Continuous Testing (Bugs & Issues Using Docker)

→ Continuous Deployment (Deployed to Production Server)

→ Continuous Feedback (Improvement of Code)

- webalizer

→ Continuous Monitoring (Release Detect System Error - Splunk)

→ Continuous Operations (Reduce planned Server Downtime - kubernetes).

These Concepts Guide the DevOps practices in each section of the lifecycle.

Q: what is Agility in the Business Process?

Business Agility refers to the company's ability to quickly adapt to changes .& fluctuations in its business environment.

The faster a Company can adjust its Business Strategy, The higher its Business Agility.

⊛ what is 3c Method in Scrum?

The 3C's (Card, Conversation, Confirmation) of User stories. work together to Come up with Ideal solutions.

The Goal is to build a shared understanding.

* What is the DevOps Lifecycle & How does IT help Businesses succeed ?

The DevOps lifecycle is an Iterative & Collaborative process that integrates automation & feedback to deliver high-quality s/w, tailored to meet business & user requirements.

This lifecycle consists of distinct phases, including planning, coding, testing, deployment, monitoring & feedback.

* Business Agility :

What is It ? & why Is it Important ?

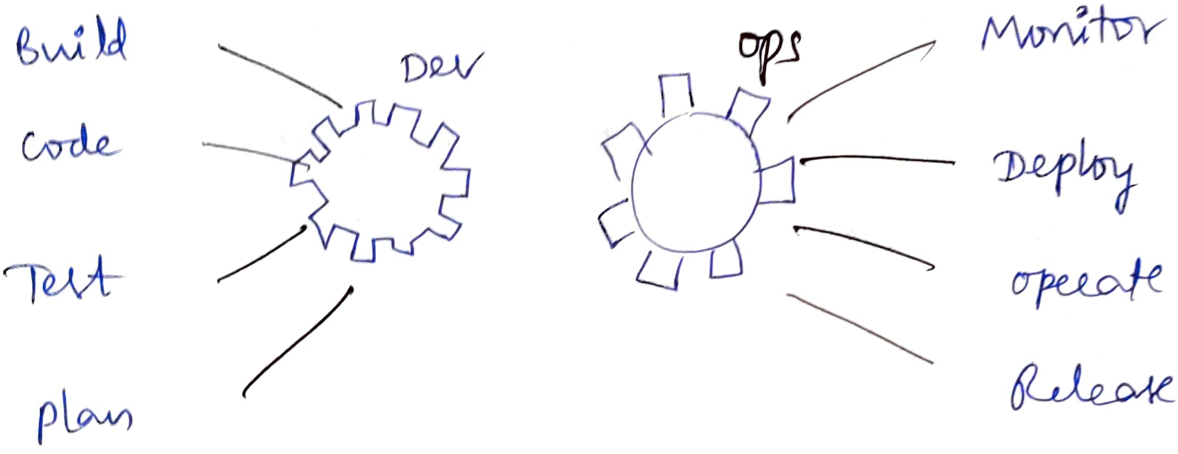
Business Agility is an Organizational Method to help businesses adapt quickly to market changes that are either external or internal.

If a Business is set up to respond rapidly & with the flexibility to meet Customer demands, they're more likely to thrive & keep those customers.

Project Mgmt s/w Can help us to keep our Ear to the ground & respond quickly to changes in the Market.

⊛ DevOps Influence on Architecture :

⊛ Devops Components:



* Introducing software Architecture :

→ It Includes the System's (Functional & Non-functional System's).

→ Flexible

→ Scalable

→ Maintainable

→ Secure.

There are no. of different s/w Arch. Styles.

→ Monolithic Architecture:

* It is the Simplest Type of s/w Architecture.

* It consists of single, large components that Contains all the System's Functionality.

→ Layered Architecture:

* It divides the system into layers, with Each layer responsible for specific set of Functionality.

→ Micro Services:

* It divides the System into Small, Independent Services that Communicate with Each other through well defined API's.

* Benefits of s/w Architecture:

→ Performance

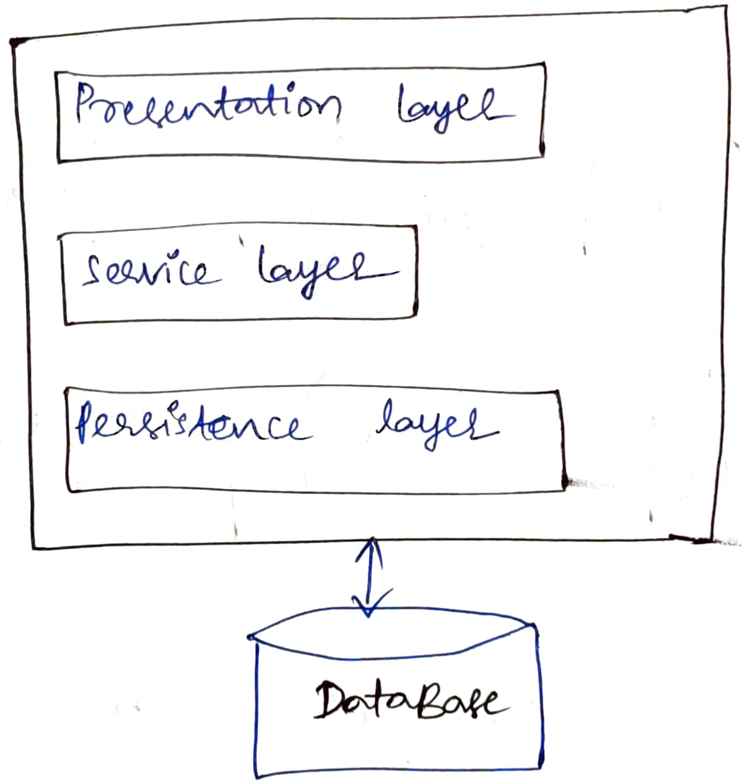
→ Scalability.

⊛ Monolithic scenario :

If All functionalities of a project exists in a single code base then, that application is Monolithic.

If we create a project, All files should be kept in a single jar/war file.

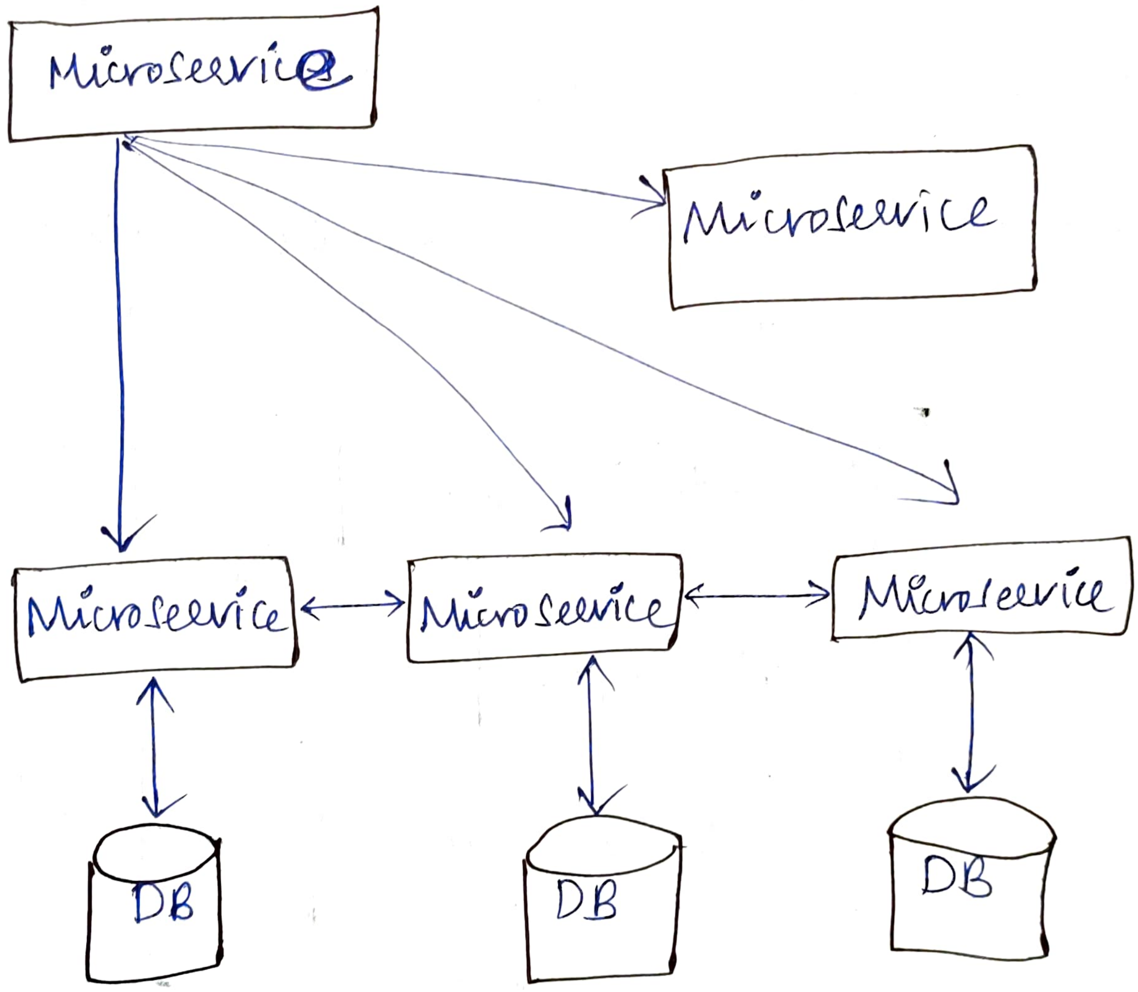
"Mono" → single code containing All required functionalities.



* MicroServices :

→ It is an Architecture Development style in which the application is made up of smaller services.

→ There services handle a small portion of functionality & data by communicating with Each other directly using protocol like HTTP.



Microservices Architecture

* Architecture ~~of~~ Rules of Thumb :

→ Use Common Sense

→ Keep it Simple

→ Assumption is the Mother of All failures.

→ Devil is in the Details

→ History is Recursively Evolving.

→ Each Advantage has a DisAdvantage

→ Make Scope Territory

→ Don't Re Invent the wheel.

→ Do the Right Things.

* Separation of Concerns:

→ It is an Important Design & Architectural Principle.

→ Every Element of a s/w Application is A Component, A layer, A package, A class or method should have the Concern & Implement it well.

* Handling Database Migrations :

① Understanding the Source DB.

② Accessing the Data.

③ Converting DB Schema

④ Testing the Migration Build

⑤ Executing the Migration

* 3 - Tier :→

3- Tier Architecture is a well established s/w Appl. organized into three logical & physical

In DevOps workflows, Database Migration Can be managed using techniques like Continuous Integration & Continuous Delivery (CI|CD) Pipelines for automated & reliable deployments, Infrastructure as code (IaC) for Consistent Environment setup, version Control for tracking changes, & automated testing to ensure data.

⊛ What Techniques can be used for database Migration in DevOps workflows?

1. Version control
2. Schema Migration Tools
3. Data Migration Tools
4. Testing strategies

Database Migration is the process of transferring data & schema from one DB System to another, often as part of a DevOps workflow that involves Continuous Integration & Delivery of web applications.

Database Migration Can be challenging, especially when dealing with large, complex, or legacy databases, or when the target database has a different structure or features than the source database.

3-Tier Architecture Includes,

- Presentation Tier (web) ↓ (HTML, CSS, JS)
- Application Tier (Processing)
- Data Tier (DB data access or backend store & Managed)

RDBMS	NO SQL
MySQL	MongoDB
Oracle	Cassandra
Sqlite	Couch DB
Sql	

⊛ DevOps Architecture :→

- Dev Team
  - ⇒ Build
  - ⇒ Code
  - ⇒ Test
  - ⇒ plan
- Ops Team
  - ⇒ monitor
  - ⇒ Deploy
  - ⇒ Operate
  - ⇒ Release

Resilence :

Resilence in DevOps refers to the ability of a software application to continue operating effectively even ~~A~~ when some components fail.

This Article discusses the concept of S/w Resilence, strategies for Achieving it, & specific suggestions for developers, DevOps, & Managers.

What is Resilence in Software Development?

Software Resilence, ~~At~~ its core, refers to the ability of a s/w system to withstand & Recover from failures, disruptions, or unexpected events.

It Encompasses a range of characteristics that enable s/w to maintain its functionality & Integrity even under challenging Conditions.

Types of Resilence :

→ Physical Resilence

→ Mental Resilence

→ Emotional Resilence

→ Social Resilence.

→ Resilence means the Capability to Recover quickly from the difficulties.

→ Resilent DevOps Team are Constantly Improving, Evolving, & finding Better way to resolve organizational challenges.

→ This Resilence allows Organization to achieve greater reliability, Continuity & Reaping the benefits of faster release cycle.

Operations ⇄ Development ⇄ QA

⊛ Resilent DevOps Teams Include,

* Adaptive

→ It is able to Pivot whenever necessary & depending upon both Internal & External pressures.

* Predictable

→ It is able to consistently & quickly respond to failure states.