—: Introduction :—

Devops :-

The Devops is the combination of two words one is development and other is operations

What is Devops?

- Devops is a software development approach that combines software development (Dev) with information technology operations (ops)
- Devops promotes collaboration between development and operations team to deploy code to production faster in an automated and repeatable way

Why Devops?

- speed up delivery
- Improve quality
- Enhance collaboration
- Increase stability
- Reduce costs.

History :-

The term "Devops" is coined by patrick Debois and Andrew clay shafer at a conference in 2009

Devops has become a mainstream approach in software development, with companies across various industries adopting its principles to accelarate delivery, improve quality etc.

=> Advantages / Benefits of DevOps

- Faster, better product delivery
- Faster issue resolution and reduced complexity
- Greater scalability and availability
- More stable operating environments
- Better resource utilization
- Greater automation
- Greater visibility into system outcomes
- Greater innovation

=> Disadvantages / cons of DevOps

- Difficult to specify design vulnerabilities
- No early phases documentation
- Security concerns
- initial implementation complexity
- Overemphasis on speed can lead to errors

=> Applications: —

- Continuous integration / continuous Deployment
- Infrastructure as code
- Automated testing
- collaboration and communication
- monitoring and logging
- Release Management
- scalability and Reliability

SDLC (Software Development Life Cycle)

The Software Development life cycle, or SDLC is a structured approach that software developers use to create, maintain, and enhance software systems. It consist of several phases, each with its own set of activities and goals.

1) Requirements Gathering:-
In this phase, developers work closely with stakeholders to understand their needs and gather requirements for the software.

2) Design:-
Once the requirements are clear, developers create a detailed design plan for the software, including its architecture, user interface, and database structure.

3) Development:-
This is where the actual coding and programming takes place, turning the design into a working software product.

4) Testing:-
The software is thoroughly tested to ensure it functions correctly, meets the requirements, and is free from bugs or errors.

5) Deployment:-
The Software is deployed or released to the users or customers, making it available for use.

The SDLC helps ensure that software development projects are well-organized, efficient, and produce high-quality software that meets the needs of users. It's like a roadmap that guides the entire software development process from start to finish.

⇒ Waterfall Model :—

The waterfall model is a traditional software development approach that follows a linear and sequential process. It's called "waterfall" because the progress flows downwards, like a waterfall.

1. Requirements Gathering:- In this phase, the project team works closely with stakeholders to gather and document all the requirements for the software. This includes understanding the needs of the users, functional requirements and any constraints or limitations

2. Design:- Once the requirements are clear, the design phase begins. The team creates a detailed design plan that outlines the software's architecture, user interface, database structure and other technical specifications. This phase focuses on planning how the software will be built.

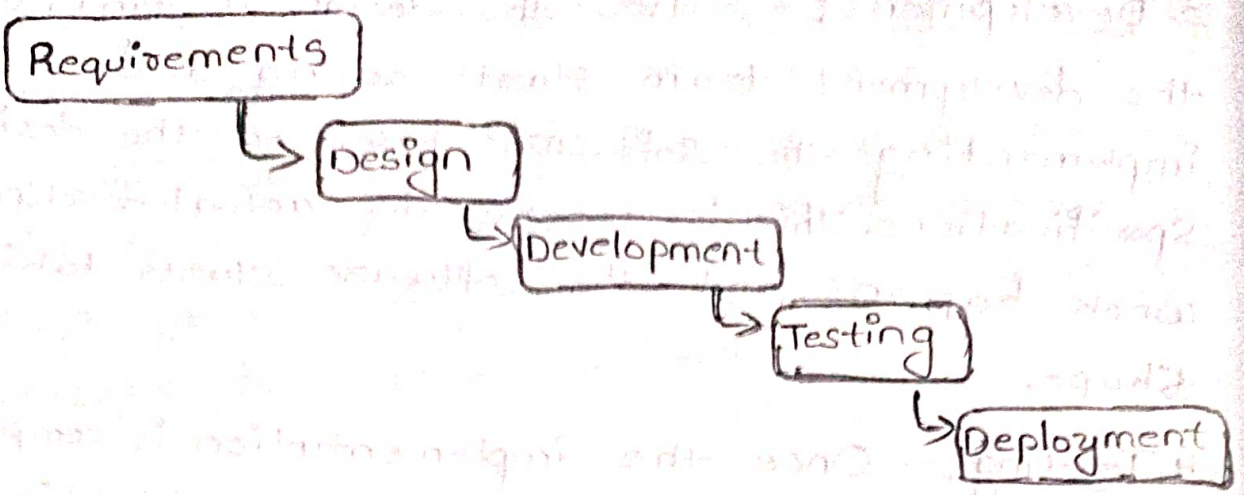
3. Development:- After the design is finalized, the development team starts coding and implementing the software based on the design specifications. This is where the actual development work happens, and the software starts taking shape.

4. Testing:- Once the implementation is complete, the software goes through rigorous testing. Testers check if the software meets the requirements, functions correctly, and is free from bugs or errors. This phase ensures the quality and reliability of the software.

5. Deployment:- After successful testing, the software is deployed or released to the users or customers. It becomes available for use, and users can start using the software to fulfill their needs.

The Waterfall model is best suited for projects with well-defined and stable requirements, where changes are less likely to occur. It provides a structured and systematic approach to software development.

- Waterfall model is lengthy and time taking process to avoid this we follow Agile methodology.



Waterfall Model

→ Advantages :-

- Easy to understand
- Individual processing
- properly Defined
- clear Milestones
- properly Documented
- Reinforces Good habits
- Working

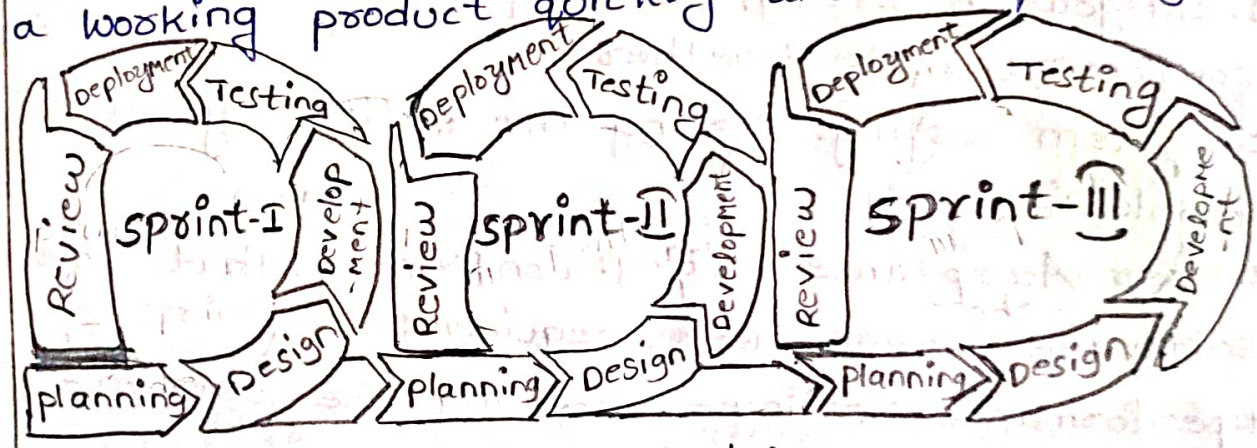
→ Disadvantages :-

- No Feedback path
- Difficult to accommodate change Requests
- No Overlapping of phases
- Limited Flexibility
- Not Suitable for Complex projects
- Late defect Detection
- lengthy development cycle.

⇒ Agile Methodology:- (Iterative approach)

Agile methodology is a project management approach that emphasizes flexibility and collaboration. It's great for adapting to change and delivering high-quality work in short iterations.

- Agile Software Development is an iterative and incremental approach to software development that emphasizes the importance of delivering a working product quickly and frequently



Life cycle of Agile methodology

→ Stages of Agile methodology:-

1. Requirements Gathering/Planning:-

- In this stage, developers work closely with stakeholders to understand their needs and gather requirements for the software
- Establishing a budget and schedule.
- Creating a project plan and allocating resources.

2. Design:-

- Developing a high-level system architecture.

- Creating detailed specifications, which include data structures, algorithms, and interfaces.
- planning for the software's user interface.

3) Development (coding)

writing the actual code for the software.
conducting unit testing to verify the functionality of individual components.

4) Testing:-

This stage involves several types of testing:

1. Integration Testing: Ensuring that different components work together
2. System Testing: Testing the entire system as a whole.
3. User Acceptance Testing: confirming that the software meets user requirements.
4. performance Testing:- Assessing the system's speed, scalability and stability.

5) Deployment

- Deploying the software to a production environment
- put the software into the real world where people can use it.
- Make sure it works smoothly in the real world
- providing training and support for end-users

6) Review (Maintenance)

- Addressing & resolving any issues that may arise after deployment
- Releasing updates & patches to enhance the software and address problems.

→ principles of Agile methodology:-

1) customer satisfaction
2) changing Requirements
3) Frequency delivery
4) promoting collaboration
5) Motivated individuals
6) face to face interaction
7) Maintain a constant pace
8) Measure progress
9) Technical excellence
10) Simplicity
11) self authorized teams
12) continuous improvements

→ Advantages of Agile methodology:-

- Immediate Feedback: It allows immediate feedback which aids software improvement in the next increment
- Adapts to changing Requirements: It is a highly adaptable methodology in which rapidly changing requirements, allowing responsive adjustments.
- Face-to-face communication: Agile methodology encourages effective face-to-face communication.
- Time-Efficient: It is well-suited for its time-efficient practices, which help in delivering software quickly and reducing time-to-market.
- Frequent changes: It effectively manages and accomodates frequent changes in project requirements according to stakeholder convenience.

6. Customer satisfaction:- It provides priority to customer satisfaction

⇒ Limitations of Agile methodology

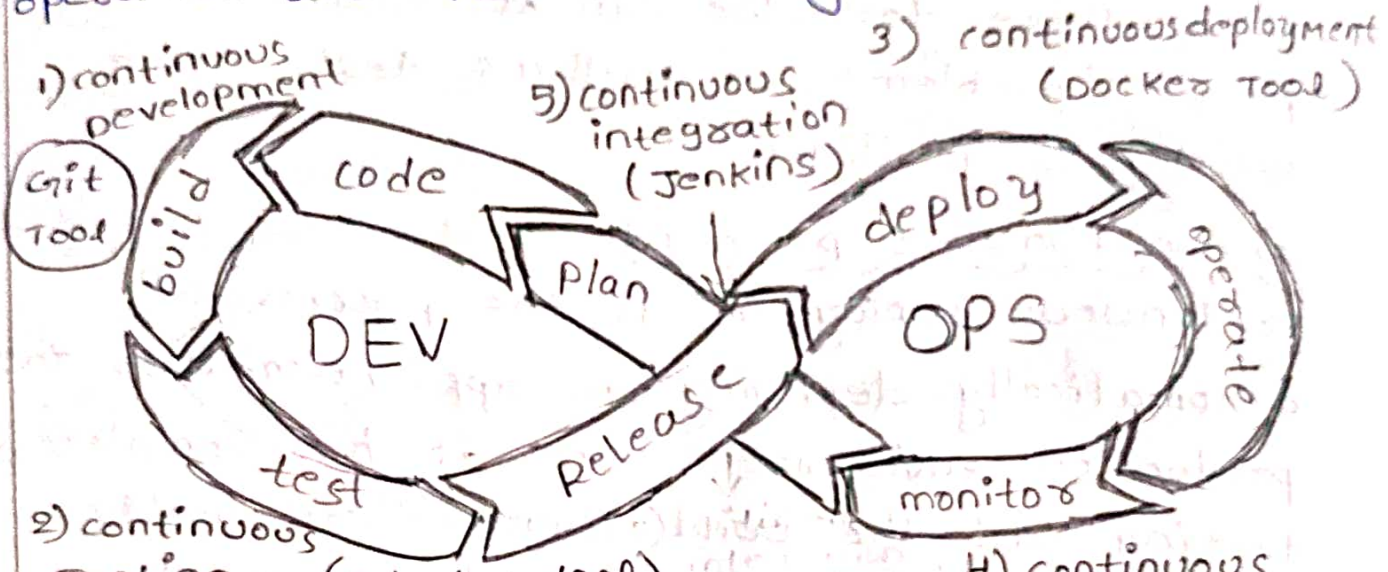
1. Less Documentation
2. challenges in large Organizations
3. Need for senior programmers
4. Limited scope control
5. predictability

⇒ popular Agile Tools for Software Development

1. Jira
2. clickUP
3. Mural
4. Kanbanize
5. GitHub

Life cycle of Devops:-

Devops lifecycle is a continuous process where development and operations teams work together to efficiently deliver software or services, involving planning, development, testing, deployment, operation and feedback stages.



-> stages of devops lifecycle:-

1. Continuous Development
2. continuous testing
3. Continuous Deployment
4. Continuous monitoring
5. Continuous integration

1. Continuous Development:-

This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. And the developers begin developing the code for the application.

- continuous development is also called as version control.

2) Continuous Testing:-
Any firm can deploy continuous testing with the use of the agile & Devops methodologies. Depending on our needs, we can perform continuous testing using automation testing tools such as Testsigma, selenium etc. with these tools, we can test our code and prevent problems, as well as test more quickly & intelligently.

3) Continuous Deployment
continuous Deployment is the process of automatically deploying an application into the production environment when it has completed testing and the build stages. Here, we'll automate everything from obtaining the application's source code to deploy it.

4) continuous monitoring
Devops lifecycle is incomplete if there was no continuous monitoring. continuous monitoring can be achieved with the help of prometheus and Grafana we can continuously monitor and can get notified before anything goes wrong with the help of prometheus we can gather many performance measures including cpu & memory utilization, network traffic, error rates and others. Grafana makes it possible to visually represent data from time series, such as cpu and memory utilization.

5) continuous integration:-

continuous integration is the most important stage of the Devops lifecycle. At this phase, updated code or new functionality & features are developed and incorporated into the existing code. In addition, defects are spotted and recognized in the code at each level of unit testing during this phase, and the source code is updated accordingly. This stage transforms integration into a continuous process in which code is tested before each commit.

-> different phases of the Devops life cycle

1) plan
2) code
3) Build
4) test
5) Release
6) deploy
7) operate
8) Monitor

23/04/2024

ITIL (Information technology infrastructure Library)

- ITIL is a frame work
- It's help for Deliver IT services.
- Help to planning, Implement & Measure services
- Develop cost effective
- Develop better quality of services
- Help to improve decision Making
- Increase productivity

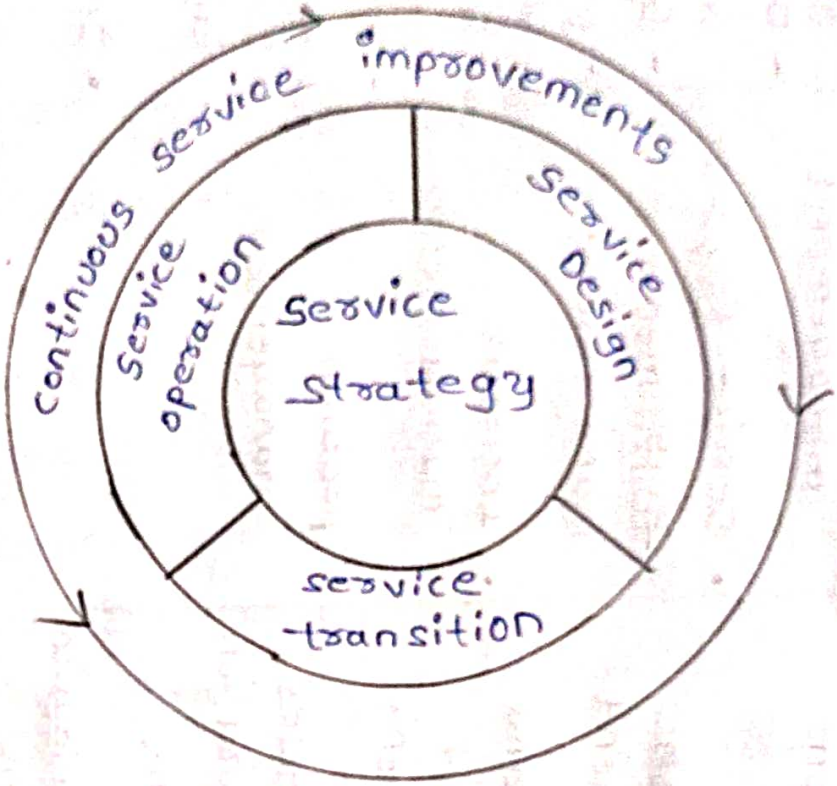
Best practice of ITIL

V1 → 1989
V2 → 1995
V3 → 2007
V4 → 2011

→ principles of ITIL :-

1) Delivery Maximum value to customers.
2) optimizing resources & capabilities
3) offering services that are useful and reliable
4) planning processes with specific goals in mind
5) Defining roles clearly for each task.

life cycle of ITIL



ITIL have 5 stages they are:-

1) Service Strategy
2) Service design
3) Service transition
4) Service Operation
5) Continuous service Improvements

These 5 stages cover 26 processes.

- Each process have activities and sub processes.

Service Strategies	Service Design	Service Transition	Service operation	continuous service Improvements
• Strategy management • Demand management • Service portfolio management • Financial management • Business Relation Management	• Service catalogue management • Availability management • Information security management • Service level management • capacity management • Design management • Supplier management • IT Server management	• Transition planning and support management • change management • Release and Deployment management • service Asses and configuration management • Service validation management • knowledge management	• Access management • Event management • Service Request Fullfillment management • Incident management • problem management	• seven step - Identifying strategy for improvement - Defining what you measure - Gathering data - processing data - Analysing information and data - processing and using information - Implementing improvement • Improvement

SERVICE STRATEGIES:-

→ Strategy Management:-

• Activities comes under strategy management

1) Identification of opportunities
2) Identification of constraints
3) organisational positioning
4) planning
5) execution

• processes comes under strategy management

1) Strategic service assessment
2) Service Strategy definition
3) Service Strategy execution

→ Demand Management:-

1) Demand prediction (prognosis)
2) Demand control.

• Activities:-

1) Analysing current usage of IT services
2) Influencing Consumption by technical or financial means
3) Anticipate the future demands

→ Service portfolio management:-

• Activities:-

1) providing live services
2) providing Retired services
3) providing service pipelines

• Sub processes comes under service portfolio mgmt:-

1) define and analyse the new services
2) Approve the changes
3) service portfolio review

→ Financial Management :—

• Activities :-

1) Accounting
2) Budgeting
3) charging

• Sub processes comes under financial management:-

1) Financial management support
2) Financial planning
3) Financial analysis and reporting
4) Service invoicing

→ Business Relationship Management :—

• Activities :-

1) Representing the service provider to the customer in a positive model
2) Identifies the business needs of a customer
3) It acts like a mediator between service provider and customer

• Sub processes comes under Business Relationship management

1) Maintaining customer relationships
2) Identify service requirements
3) sign-up customers to standard services
4) customer satisfaction survey

5) Handles customer complaints
6) Monitor customer complaints.

⇒ SERVICE DESIGN

-> Service catalogue management (SCM)

Allows the catalogue manner to give information about all other management processes.

Types:-

- BSC or Business service catalogue
- TSC or Technical Service catalogue

-> Availability management:-

Responsible for ensuring that the services of IT meet the agreed availability goals.

• Activities.

* Reactive Activity
* proactive Activity

• subprocesses comes under Availability management

* Design IT services for availability
* availability testing
* availability monitoring and Reporting

-> Information Security management:-

Main aim is to verify the confidentiality, integrity and availability of data, information and services of IT organization

• Activities

* plan
* implement
* Evaluation
* Maintain

• Subprocesses comes under Information security Mgmt:

* Design of security controls
* Validation and testing of security
* Management of security incidents
* Security review

→ Service level Management :—

Service level management is mainly responsible for gathering service requirements, as well as monitoring and reporting with regards to agreed service levels.

• Subprocesses of service level management

* Maintenance of the SLM framework
* Identification of service Requirements
* Agreements sign-off and service Activation
* Service level monitoring and Reporting

→ Capacity Management :—

Ensures that the capacity of IT service can meet the agreed capacity in a cost-effective and timely manner.

Activities :—

* Manages performance of resources so that IT Services can easily meet SLA targets.
* Creates & Manages capacity plan which aligns with strategic plan of an organization
* reviews performance of service & capacity of current service periodically.
* Understands current and future demands of customer for IT resources.

Subprocesses comes under capacity management

* Business capacity management
* service capacity management
* Component capacity management
* capacity management Reporting

-> Design Management :-

. Design Management involves designing new or changed services to meet business and customer needs.

. This process includes all design activity, particularly all new or changed service solutions

-> Supplier Management :-

Here the supplier manager is responsible for verifying that all supplies meet their contractual commitments.

Activities :

* Manages subcontracted suppliers
* Manages relationship with suppliers
* helps in implementing supplier policy
* Manages supplier policy & supports scmis
* Manages & maintains the performance of suppliers

Subprocesses comes under Supplier management

* provides framework of Supplier management
* selection of new contracts & Suppliers
* establish new contracts & suppliers
* process the standard orders
* contract & Supplier review
* Contract renewal or Termination.

→ IT server Management :-

- IT server management is also called as IT service continuity Management
- Allows the continuity manager to maintain risks which could impact on service of IT

Activities :-

* Initiation
* Requirements & strategy
* Implementation
* Ongoing operation

⇒ SERVICE TRANSITION

→ Transition planning & support management:-

In this process, the project manager plays a role as an Owner. This management process helps the service transition projects. sometimes this process is also known as the project management process.

Activities :-

* It manages the issues & risks
* It defines the task and activities which are to be performed by the separate processes
* It makes group with the same type of releases
* It manages each individual deployment as a separate project.

Subprocesses comes under Transition planning & support management

* Initiate the project
* planning and coordination of a project
* project control
* project communication and Reporting.

→ Change Management :-

. In this process the change manager plays a role as an Owner. The change manager controls or manages the service lifecycle of all changes.

. It also allows the change manager to implement all the essential changes to be required with the less disruption of IT services.

. This management process also allows its owner to recognize & stop any unintended change. It is tightly bound with the "service Asset and configuration management".

⇒ Types of changes:

1) Normal change
2) Standard change
3) Emergency change

Subprocesses comes under change management

* change Management support
* RFC (Request for change) logging and Review
* change Assessment by the owner
* Assess and implement the emergency changes
* Assessment of change proposals
* change scheduling and planning

* change development Authorization
* implementation or deployment of change
* minor change Deployment

=> Release and Deployment Management :-

- In this process, the release manager plays a role as an owner. Sometimes, this process is also known as the 'ITIL Release Management process'.
- This process allows the release manager for managing, planning and controlling the updates & release of IT services to the real environment

=> Types of releases defined by ITIL :-

1) Minor release
2) Major release
3) Emergency release

subprocesses comes under release and deployment management

* Release management support
* Release planning
* Release Build
* Release Deployment
* Early life support
* Release closure.

→ Service Asses and configuration management :-

. In this process, the configuration manager plays a role as an owner.

. This management process is a combination of two implicit processes

1) Asset Management

2) Configuration management.

. The Aim of this management process is to manage the information about the configuration items. which are needed to deliver the services of IT

→ Sub processes :-

* planning and management
* configuration control and identification
* status Accounting and reporting
* Audit and Verification
* Manage the information.

→ Service Validation management :-

. In this process, the test manager plays a role as an Owner. The main goal of this management process is that it verifies whether the deployment releases and the resulting IT services meet the customer expectation.

. It provides the quality assurance for both the services and components.

Activities

* Validation and test management
* planning and Design

* Verification of test plan and design
* preparation of the test environment
* Testing
* evaluate exit criteria and Report
* clean up and closure.

Subprocesses comes under service validation

* Test model definition
* Release component Acquisition
* Release Test
* Service Acceptance testing.

-> Knowledge Management :—

. In this process, the knowledge manager plays as an owner. This management process helps the knowledge manager by analysing, storing & sharing the knowledge and the data or information in an entire IT organization

. under this knowledge management processes, no sub-process is specified or defined.

SERVICE OPERATION :—

-> Access Management :—

Access Manager plays a role as an owner. This type of Management process is also Sometimes called as the 'Identity Management' or 'Rights Management'.

Activities :-

* Request Access
* Verification
* providing Rights
* monitoring or observing the identity status
* logging and tracking status
* Restricting or Removing Rights

Sob-processes :-

* Maintenance of catalogue of user Roles and Access profiles
* processing of user Access Requests.

→ Event Management :-

- It allows the IT operations Manager to decide the appropriate action for the events.
- It also provides the trigger for the execution of management activities of many services.

The Event monitoring Tools are divided into 2 types

1. Active Monitoring Tool
2. passive Monitoring Tool

Following are the three types of events which are defined by the ITIL :-

1. Warning
2. Information
3. Exception.

Sub-processes:-

* Event monitoring and Notification
* First level correlation and event filtering
* Second level correlation and Response selection
* Event Review and closure

→ service Request fullfillment management:-

• service Request fullfillment management is the process of handling service requests from users.

Activities:-

* Request logging
* classification and categorization
* prioritization
* Approval
* fulfillment
* monitoring & tracking

subprocesses:-

* Request intake
* Request classification
* Request prioritization
* Request Approval
* Request communication
* Request closure & Documentation.

→ Incident management :-

Incident Manager plays a role as an Owner. the main goal of this management process is to maintain or manage the life cycle of all the incidents which happen in the services of IT.

sub-processes :-

* Incident Management support
* Incident Logging and categorization
* pro-active user information
* First level support for Immediate incident Resolution
* Second level support for Incident Resolution
* Handling of Major incidents
* Incident monitoring & Escalation
* closure and Evaluation of incident
* Management Reporting of incident

→ problem Management :-

The main goal of this management process is to maintain or manage the life cycle of all the problems which happen in the services of IT.

Activities :-

* problem Detection
* problem logging
* categorization of a problem
* prioritization of a problem
* Investigation & diagnosis of a problem
* Identify workaround

* Raising a known Error Record
* Resolution of a problem
* problem closure
* Major problem Review.

=> Continuous service Improvements :-

- seven steps

1) Identifying strategy for improvement:-

Two important Questions

1) what we are trying to achieve?
2) How can improvements enable the business vision to be achieved

Inputs:-

1) Budget cycle
2) legislative
3) Governance
4) Balanced score card
5) service catalogue

outputs:-

- vision statement
- mission statement
- objectives
- Goals

2) Defining what you Measure :-

-> what do you actually Measure

A) Listing the tools currently you have

-> where will you find information

A) Each process, procedure, work instruction existing reports databases

Inputs :-

SLR and targets

service review meeting

outputs :-

Measurable metrics

Gap analysis of metrics

Measurement plan

3) Gathering Data :-

- Gather whatever data has been identified as both needed and measurable.
- Remember all the data not gathered automatically
- Manual data entry is required in SMART (specific - Measurable - Achieve - Relevant - Timely)

Inputs :-

1) New business requirements
2) Existing SLAs.
3) previous reports

Outputs :-

capacity plans, updated availability indentified tools to use collection of data agreement on the data integrity.

4) processing Data:-

- convert data into required format

Inputs:-

- SLAs, OLA
- service catalogue
- Report template
- Report frequency

outputs:-

- Reports
- updated availability
- capacity plan

5) Analysing information and data

- Data is processed as information then analysis starts

→ Are there any clear trends?
→ Are they +ve or -ve trends
→ Are changes required
→ Are we operating According to plan
→ Are we meeting targets?

Inputs:-

- Monitoring Data
- problem-solving techniques
- customer satisfaction surveys

outputs:-

- Trends
- observations
- Statistical Analysis Reports
- Bottlenecks
- Evaluation reports.

6) processing and using information:—

. utilizing information means analyzing data to enhance service delivery and IT operations.

Input:—

. collect data and information
. user inputs
. stackholder contact details

outputs:—

. Funding Requirements
. Analyzed Data
. cost - benefit analysis

7) Implementing Improvements:—

. It involves identifying areas for enhancement, planning changes and implementing them.

Inputs:—

1) Gained Approval on agreed implementation plans
2) A CSI register

outputs:—

. Action plans
. project plans
. initiatives
. change requests

→ Improvement :—

- Improvements in ITIL focus on enhancing efficiency, effectiveness, and quality of IT service management processes through continuous evaluation, adaptation & optimization.

Activities :—

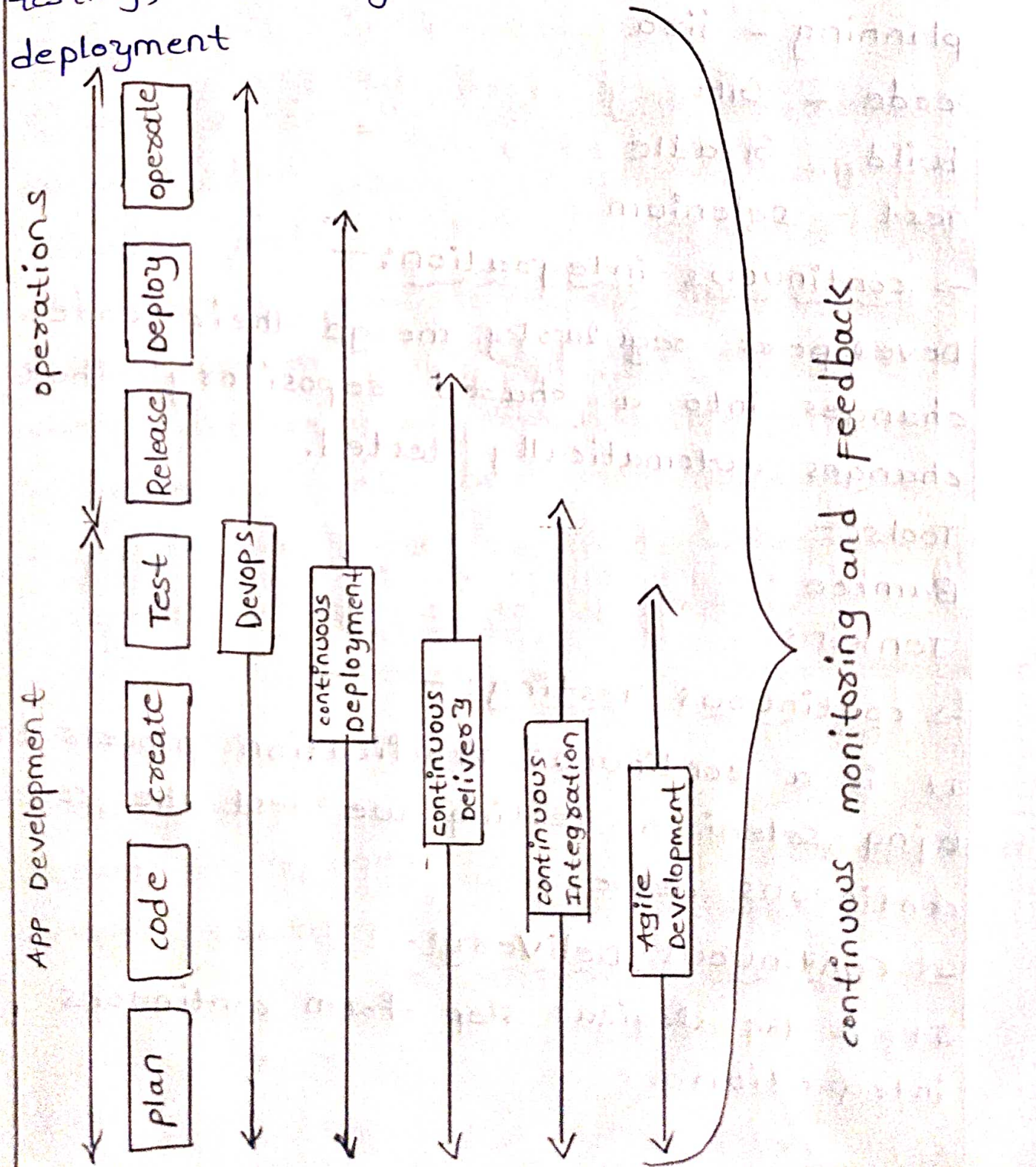
1) Assessment
2) Analysis
3) planning
4) Implementation
5) Monitoring
6) Review
7) Adaptation

Subprocesses :—

1) problem Identification
2) Root cause Analysis
3) change Management
4) Continual service improvements
5) Metrics and Reporting
6) Feedback & Review
7) knowledge management

Devops process :-

The Devops process flow is all about agility and automation. Each phase in the Devops lifecycle focuses on closing the loop between development and operations and driving production through continuous development, integration, testing, monitoring and feedback, delivery, and deployment



→ continuous development / Agile development:-

It is a iterative process. we can deliver high quality product with lower risk

- we can achieve faster delivery
- we can do changes easily with the help of automation tools

Tools:-

planning - jira

code - Git

build - Gradle

Test - selenium

→ continuous integration:-

Developers regularly merge their code changes into a shared repository that changes automatically tested.

Tools:-

Bamboo

Jenkins

→ continuous Testing:-

It is a continuous Verification process by using selenium testing we test the code continuous delivery

→ Continuous Delivery:-

It is the logical step from continuous integration

. The main goal of continuous delivery is to release updates to the users rapidly and sustainably

→ Continuous deployment:-

. It is fully automated in devops process (no human involvement)

Tool:- Docker

⇒ Scrum:-

Scrum is the type of Agile Framework within which people can address complex adaptive problem while productivity & creativity of delivering product is at highest possible values. scrum uses iterative process.

silent features of scrum are:-

- Scrum is light-weighted framework
- Scrum emphasizes self-organization
- scrum is simple to understand
- Scrum framework help the team to work together

Scrum includes group of people called as Scrum team. Normally contains 5-9 members

Scrum team:- 1) product owner
2) scrum master
3) developer
4) tester

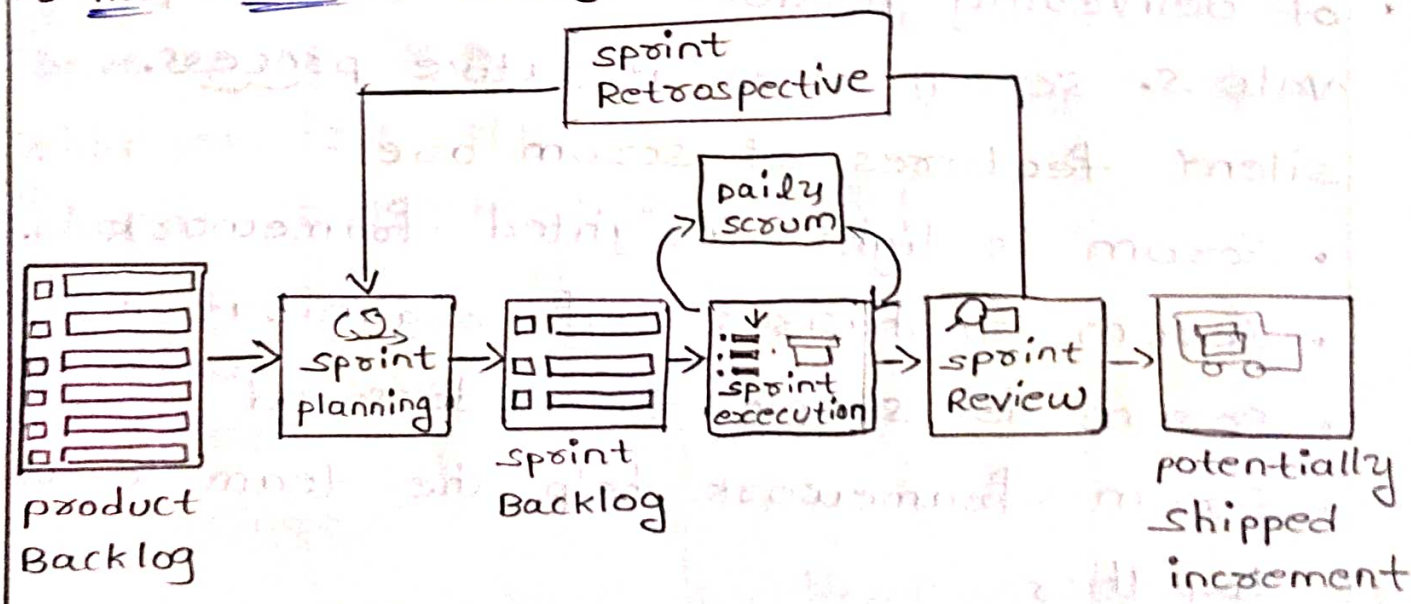
1) product owner:- They decide what needs to be done first and make sure the team understands what to build

2) scrum master:- they help the team work well together and solve any problems that come up

3) Developer:- They build the product and make sure it works properly.

4) Tester:- They check the product to find any mistakes and make sure it's good to go.

⇒ The scrum lifecycle:-



• product Backlog:- This is where all the tasks needed for the project are listed. It's like a to-do list for the team.

• sprint planning:- The team selects tasks from the backlog to work on during the next sprint, which is a short period of time

(usually 2-4 weeks).

- Sprint:- During the sprint, the team works on the selected tasks to create a piece of the product, called the increment.
- Daily scrum:- Every day, the team meets for a quick update on what they're working on and any challenges they're facing
- Sprint Review:- At the end of the sprint, the team shows what they've accomplished to stakeholders and gets feedback
- Sprint Retrospective:- The team reflects on what went well and what could be improved, so they can do better in the next sprint.

-> Task board in scrum

It is a visual representation of the work that needs to be done during a sprint. It typically consists of 3 main columns:

1) To-Do: This column contains tasks that have not yet been started.
2) In progress: This column shows tasks that team members are currently working on.
3) Done: This column displays tasks that have been completed.

Advantages of scrum framework :-

- scrum framework is fast moving & money efficient
- scrum framework works by dividing the large product into small sub-products. It's like a divide and conquer strategy
- In scrum customer satisfaction is very important
- scrum is adaptive in nature because it have short sprint.

Disadvantages of using scrum framework

- scrum framework do not allow changes into their sprint
- It can be difficult for the scrum to plan, structure and organize a project that lacks a clear definition
- The daily scrum meetings & frequent reviews require substantial resources.

Kanban :-

Kanban is a method for managing work, emphasizing just-in-time delivery and visualizing the process. It aims to identify bottlenecks and optimize workflow for efficient throughput.

Kanban principles :-

1) Visualize work :- Represent tasks on a visual board to enhance transparency
2) Limit work in progress (WIP) :- set constraints to prevent overloading team members.
3) Manage Flow :- optimize the movement of tasks through the system
4) Make process policies Explicit :- clearly define rules and guidelines for task management
5) Implement Feedback Loops :- Establish mechanisms for continuous improvement.
6) Improve collaboratively, Evolve Experimentally :- Foster a culture of teamwork and experimentation.
7) Encourage Leadership at All levels :- Empower team members to take initiative
8) Respect current Roles, Responsibilities, and Job Titles :- Honor existing structures while implementing kanban.

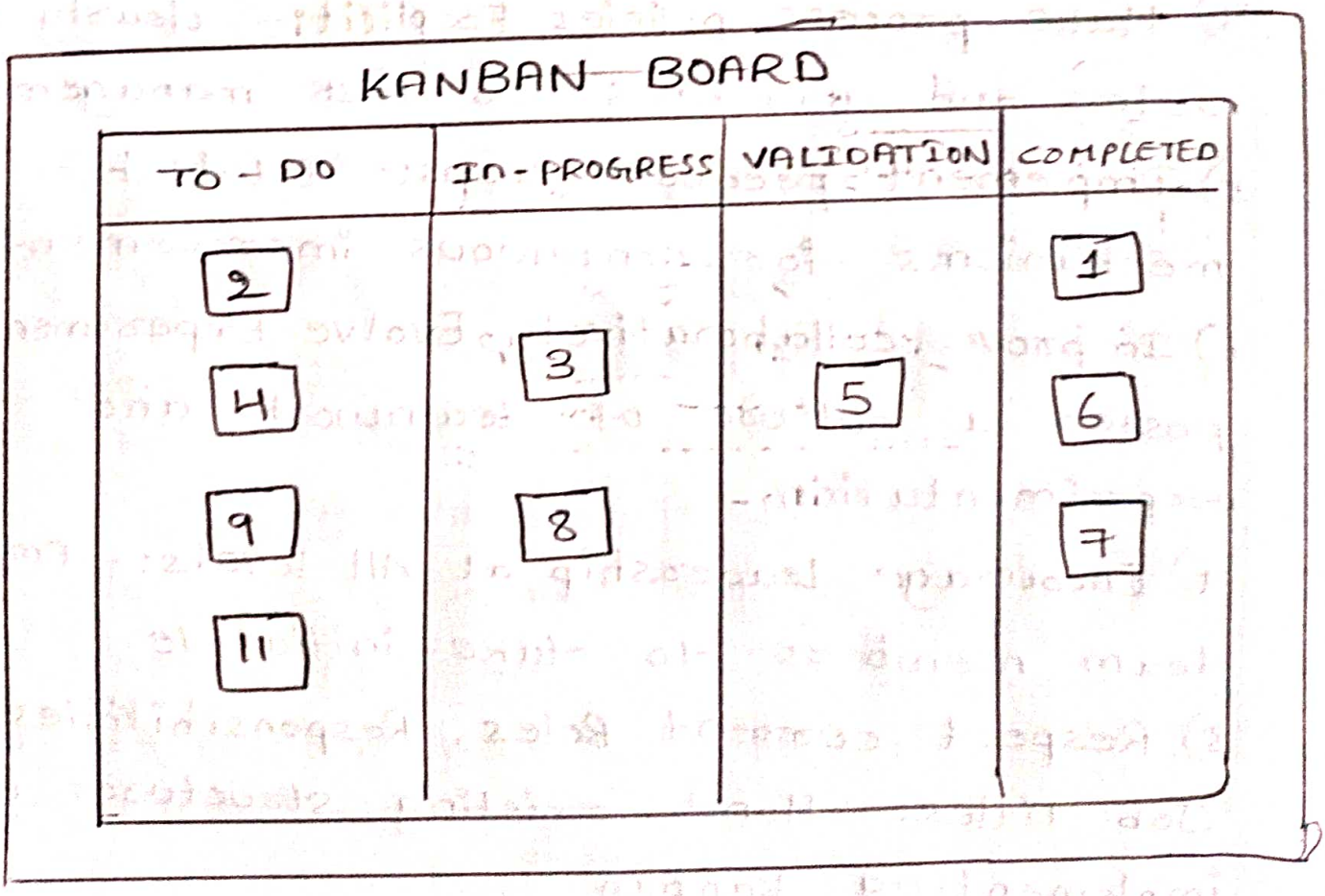
9) Encourage Acts of leadership at All levels:- promote leadership behaviors across the organization

10) Evolve organically:- Allow the kanban system to adapt naturally over time.

Kanban workflow:-

The kanban workflow typically consists of 4 phases:-

1) To Do:- Tasks awaiting assignment
2) In progress:- Tasks being worked on
3) Review/QA:- Tasks undergoing quality checks
4) Done:- completed tasks ready for development

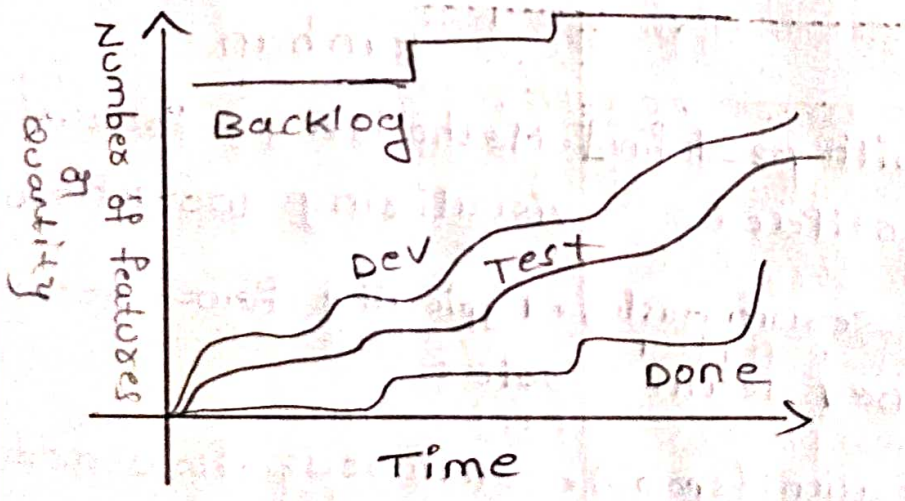


-> pull method in kanban :-

1. Tasks are pulled as capacity allows
2. WIP limits prevent overloading
3. Work progresses based on demand
4. Ensures continuous flow
5. prioritizes customer needs.

-> cummulative flow diagrams in kanban:-

- show work progression over time
- use horizontal axis for time
- Vertical axis measures work quantity
- Lines or areas represent workflow stages
- Identify bottlenecks and inefficiencies
- Guide process optimization



-> Kanban with Azure Boards:-

- Visual board for task management
- customizable workflow stages.
- work item tracking and updates
- WIP limits for flow optimization
- integrated with Azure Devops.

Advantages of Kanban :-

- Flexibility in workflow management
- Efficient resource utilization with WIP limits
- Enhanced visibility and transparency
- culture of continuous improvement
- Reduced waste & customer-focused delivery

Disadvantages of Kanban :-

- Complexity in large-scale projects
- challenges in managing dependencies
- Limited long-term planning capabilities
- potential for reduced team autonomy
- Learning curve for adoption

→ Difference between scrum and Kanban

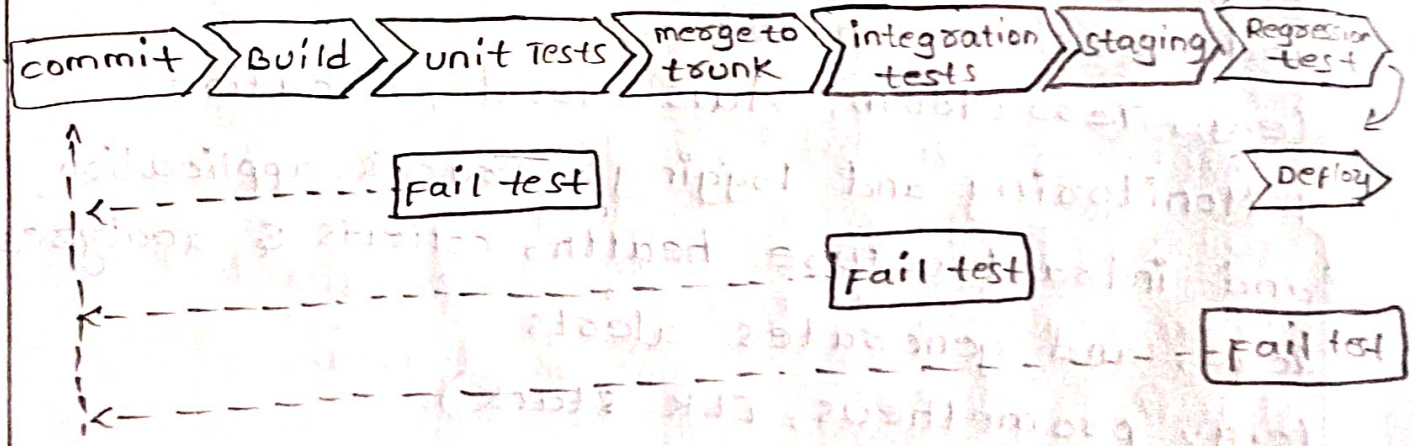
Scrum	Kanban
• Framework with predefined roles, events, artifacts	• Methodology focused on visualizing workflow
• Defined roles: Scrum master, product owner, Dev Team	• Typically fewer predefined roles
• Time-boxed iterations (sprints)	• continuous flow model
• upfront planning & commitment for each sprint	• more flexibility in planning
• sprint backlog for each iteration	• WIP limits to manage flow
• It uses review as key metric	• It uses cummulative flow diagrams
example:- It use GIT Tool	example:- It use Azure

Delivery pipeline :-

A delivery pipeline automates software delivery from development to production, with stages for code progress and checks, aiming for rapid, reliable deployments with high quality and minimal risk

considerations for building a DevOps pipeline

DevOps pipelines are tailored to each organization's tech stack and engineer skills. They typically include stages like commit, Build, Tests, integration, staging, Regression, and Deployment. pipeline stops on failure, providing feedback to developers.



components of DevOps pipeline: -

1) Version control System (VCS): Manges code changes and facilitates collaboration among team members (e.g Git, SVN)

2) Continuous integration (CI): Automates code integration, building, and testing to provide rapid feedback to developers
(e.g Jenkins, Git lab CI).

3. Artifact repository:- centralizes storage of build artifacts like binaries & Docker images (e.g, Nexus, Artifactory)

4. Automated Testing:- Executes various tests to ensure software functionality & reliability (e.g., Junit, Selenium)

5. Continuous Deployment (CD):- Automates deployment processes to rapidly & consistently deploy software changes (e.g Ansible, Kubernetes)

6. Infrastructure as code (IaC):- Defines and manages infrastructure using code for automation and version control (e.g., Terraform, Aws cloud formation)

7. Monitoring and logging:- Tracks application and infrastructure health, collects & analyzes logs, and generates alerts (e.g, prometheus, ELK stack)

8. Security and compliance:- Integrates security scans and compliance checks into the pipeline to ensure software meets standards (e.g., OWASP ZAP, SonarQube).

⇒ Bottlenecks in Devops:-

In Devops, bottlenecks can occur at various stages of the software delivery pipeline, hindering the flow of work & slowing down the release process. some bottlenecks in Devops include:

- Manual processes: Reliance on manual tasks slow down work flow.
- Limited Automation: Inadequate automation of repetitive tasks hinders efficiency
- Resource constraints: Insufficient resources impede scalability and productivity
- Dependency management:- Delays in obtaining external resources disrupt workflow
- Testing delays: Lengthy or inefficient testing processes delay feedback and release.
- Inadequate tooling:- outdated or inefficient tools hinder productivity
- Lack of visibility:- poor visibility into work status impedes decision-making
- culture & communication barriers: siloed teams and communication issues hinder collaboration and information flow.

Examples of bottlenecks in DevOps:-

1) Manual processes: Manual infrastructure provisioning instead of automation
2) Limited automation: Lack of automated testing, relying on manual checks
3) Resource constraints: Insufficient cloud resources causing deployment failures
4) Dependency Management: waiting for third-party updates before processing
5) Testing delays:- long wait times for test environment availability
6) Inadequate tooling:- Outdated CI/CD tools hindering pipeline efficiency
7) lack of visibility:- Inability to track task status, leading to delays
8) communication barriers:- Siloed teams slowing down issue resolution.