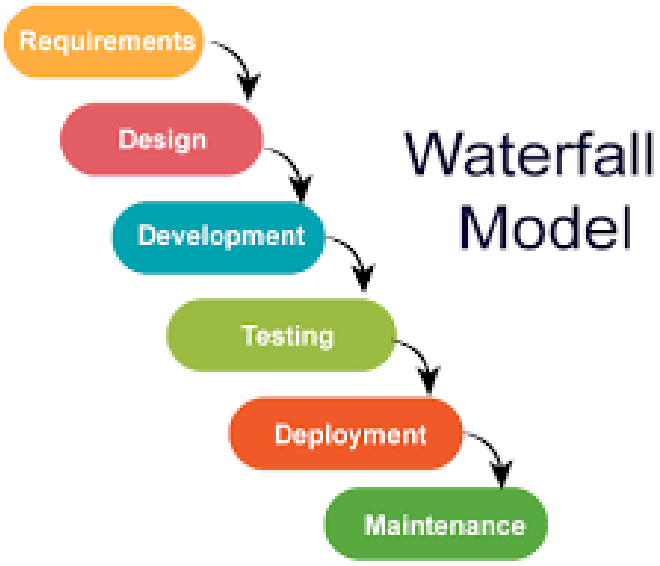
Product Management: simplify and streamline software development and helps ensure smooth delivery

- **Agile**
- **Scrum**

Waterfall Method:



Gather requirement = seniors
Designing = SA + Dev
Development = Dev
Testing = Security + QA
Deployment = Ops
Maintainance = Ops

GET Method for API :

PUT Method for API :

Agile: Small and Frequent Chnages

- All Changes are independent

- Ability to create and respond to changes

12 Principles of Agile

1. Our highest priority is to satisfy the customer through
the early and continuous delivery of valuable software.

a. Clear Understanding of Requirements

b. Delivery Qulaity

c. Repect Time Line

d. Less Costly (time and Money)

Deliverable = anything which you developed as part of app
now need approval / show it to customers

2. Welcome changing requirements,
even late in development.
Agile processes harness change for the customer's competitive advantage

3. Deliver working software frequently,
from a couple of weeks to a couple of months,
with a preference to the shorter timescale.

Sprint = 2 weeks

1. Design ~ 1 week
2. POC ~ 2 weeks
3. Work on Feedback ~ 3 days
4. Dev
5. Test
6. Deploy
7. maintain

4. Business people and developers must work together daily throughout the project.

- Ensure no communication Gap

5. Build projects around motivated individuals.

Give them the environment and support they need, and trust them to get the job done.

- No Micromanagement

6. The most efficient

and effective method of conveying information

to and within a development team is face-to-face conversation.

a. Ensures Clear Communication

b. Saves Time

7. Working software is the primary measure of progress.

- Proof of Work : Github Repo

Dummy App

8. Agile processes promote sustainable development.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity–the art of maximizing the amount of work not done–is essential.

- avoid over-engineering of task

skip docker / K8s / Serverless - blindly NO

11. The best architectures, requirements, and designs emerge from self-organizing teams.
- give facts to explain yourself

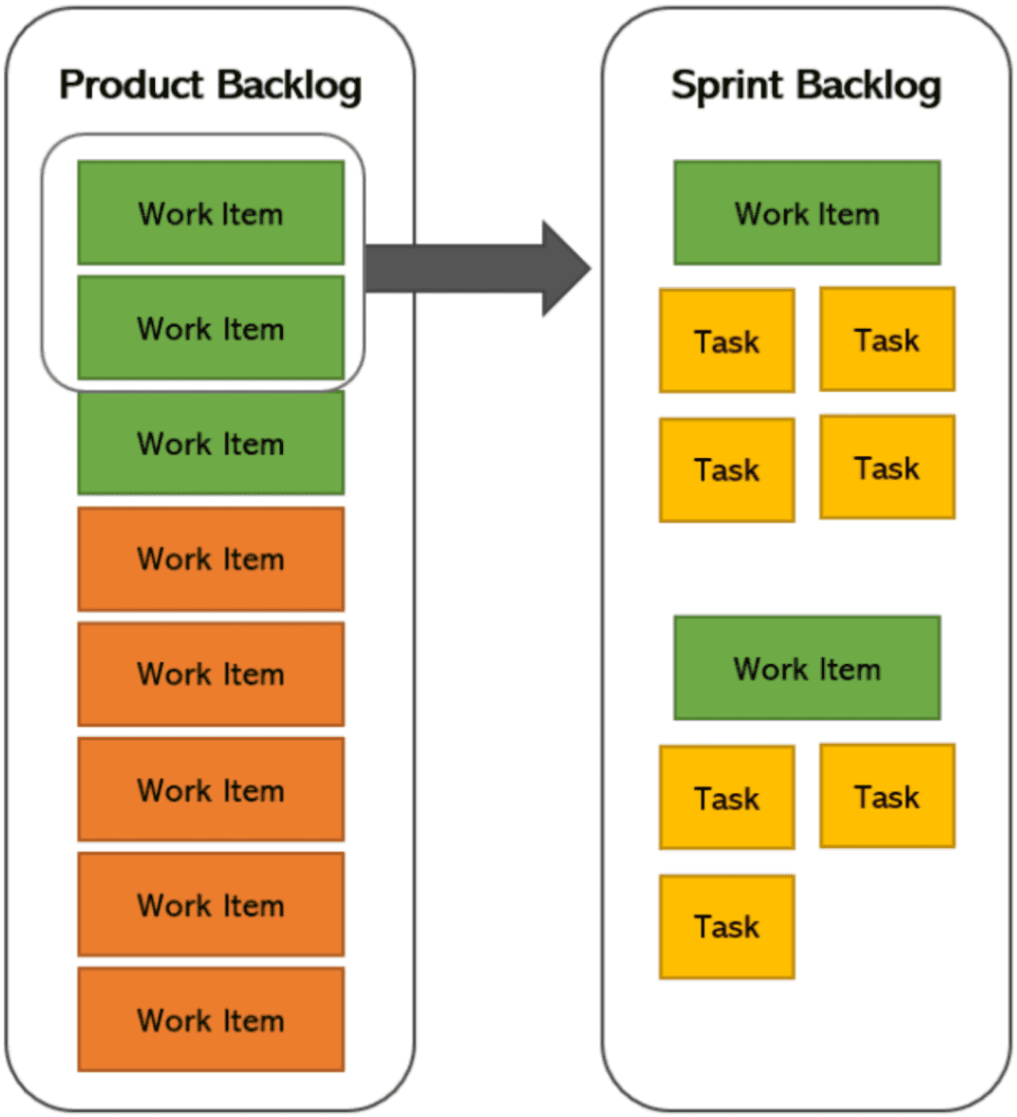
12. At regular intervals, the team reflects on how to become more effective,
then tunes and adjusts its behavior accordingly.

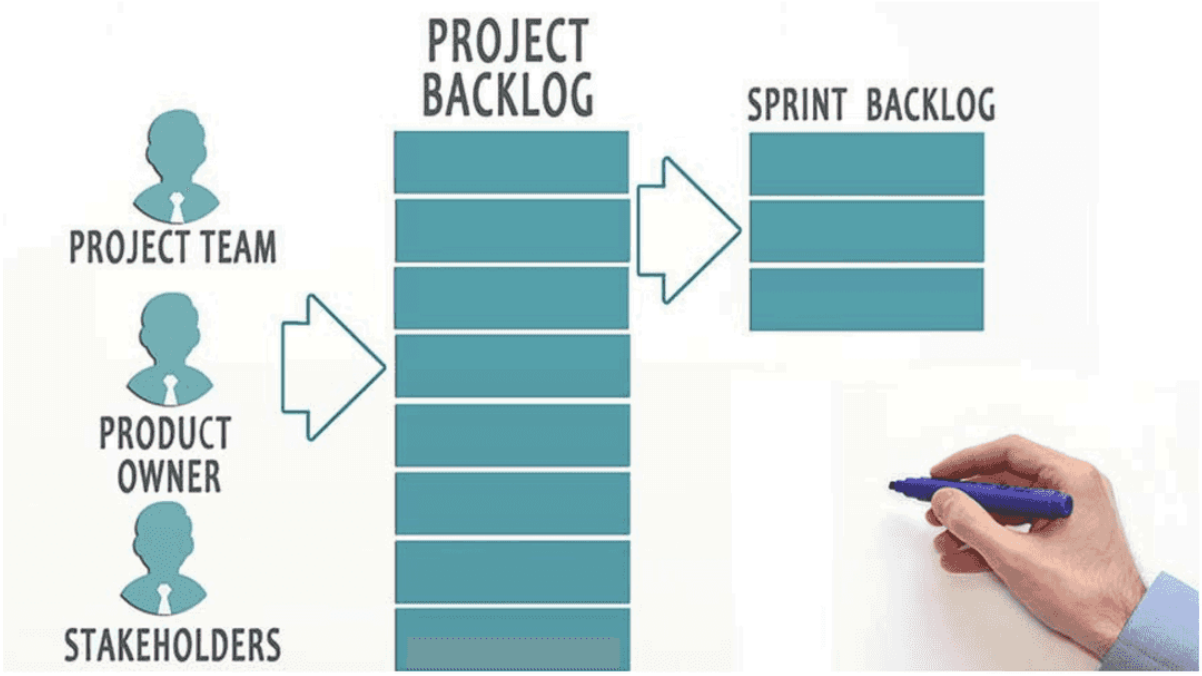
- "Anything which can't be measured
Can't be improved "

Scrum is a management framework that teams use to self-organize and
work towards a common goal.
It describes a set of meetings, tools, and roles for efficient project delivery. Much like a sports team practicing for a big match,
Scrum practices allow teams to self-manage, learn from experience, and adapt to change

Product Owner :
responsible for ensuring the success of a project in Scrum. The product owner is responsible for managing and
optimizing the product backlog in order to maximize the value of the product

A product backlog is a prioritized list of work for the
development team that is derived from the product roadmap and its requirements.
The most important items are shown at the top
of the product backlog so the team knows what to deliver first.


Product Backlog
Work Item
Work Item
Work Item
Work Item
Work Item
Work Item
Work Item
Work Item
Sprint Backlog
Work Item
Task
Task
Task
Task
Work Item
Task
Task
Task


PROJECT BACKLOG
SPRINT BACKLOG
PROJECT TEAM
PRODUCT OWNER
STAKEHOLDERS

Create WebPage

Make API that caluclates EMI

COnnect API to Webpage

Update Database with Customer Info

Project / Product Backlog = collection of Epic, user story

Time Frame = Sprints

sprint-1:
28 - 6 Sept: Website

Sprint-2:
9 sept - 23 Sept : API Part

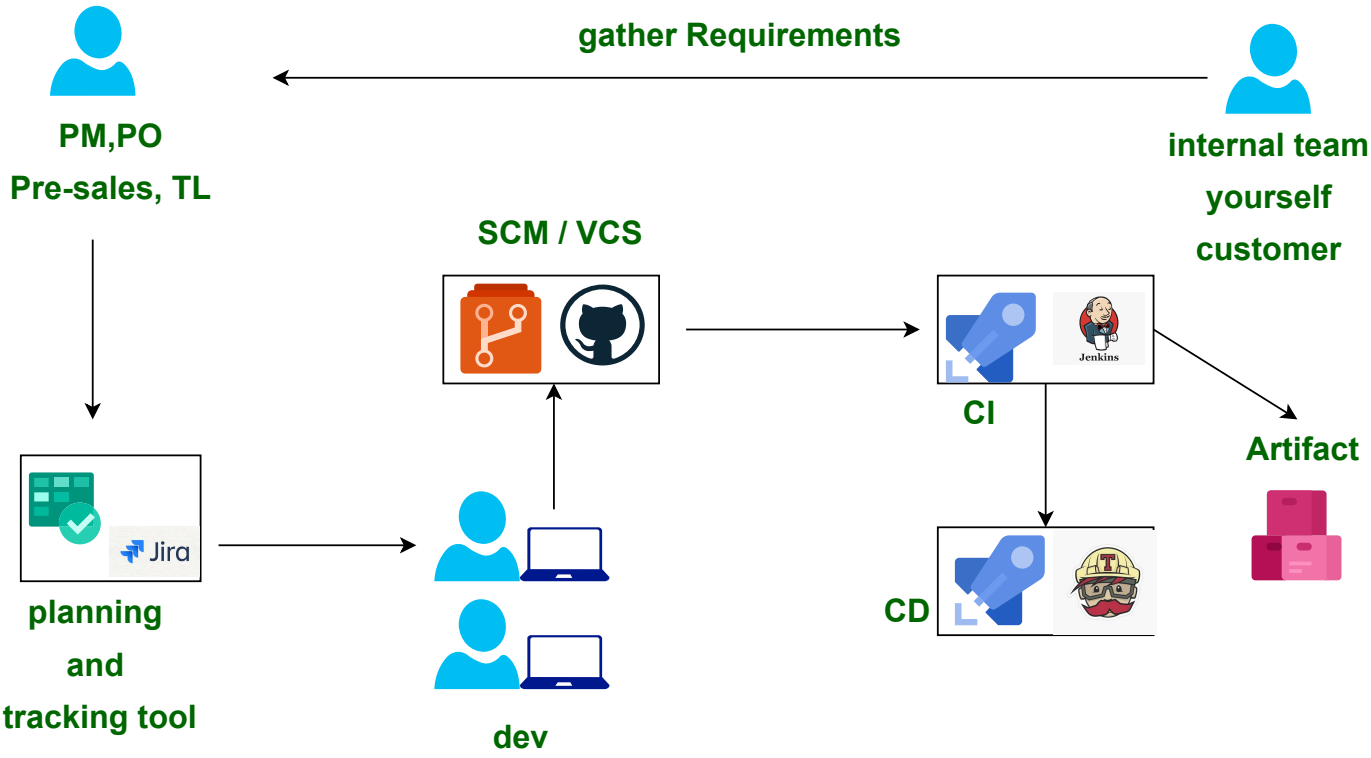
on 28 Aug Decide = Each Day , what task to perform
(Sprint Planning)
User-1: Website Creation

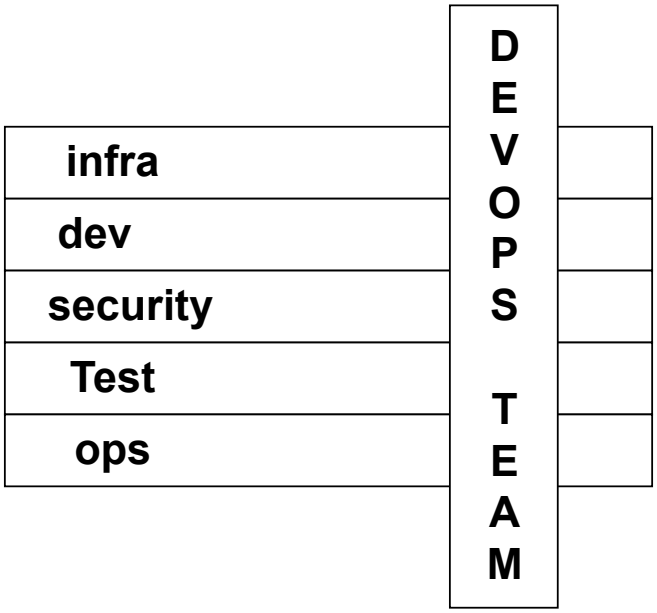
28 AUg - 6 sept :
create a VM
Frontend Ready in React JS
Backend In Python + MYSQL

sprint review meeting - once a week
Daily stand up meetings

Azure DevOps:

software Development Lifecycle [SDLC]





DevOps: methodology
where People, Proccess and Product

example : A Team of 5 people using azure devops
to deliver project using agile methodology

- Automate every repeated task
- continues monitoring
- work on feedback

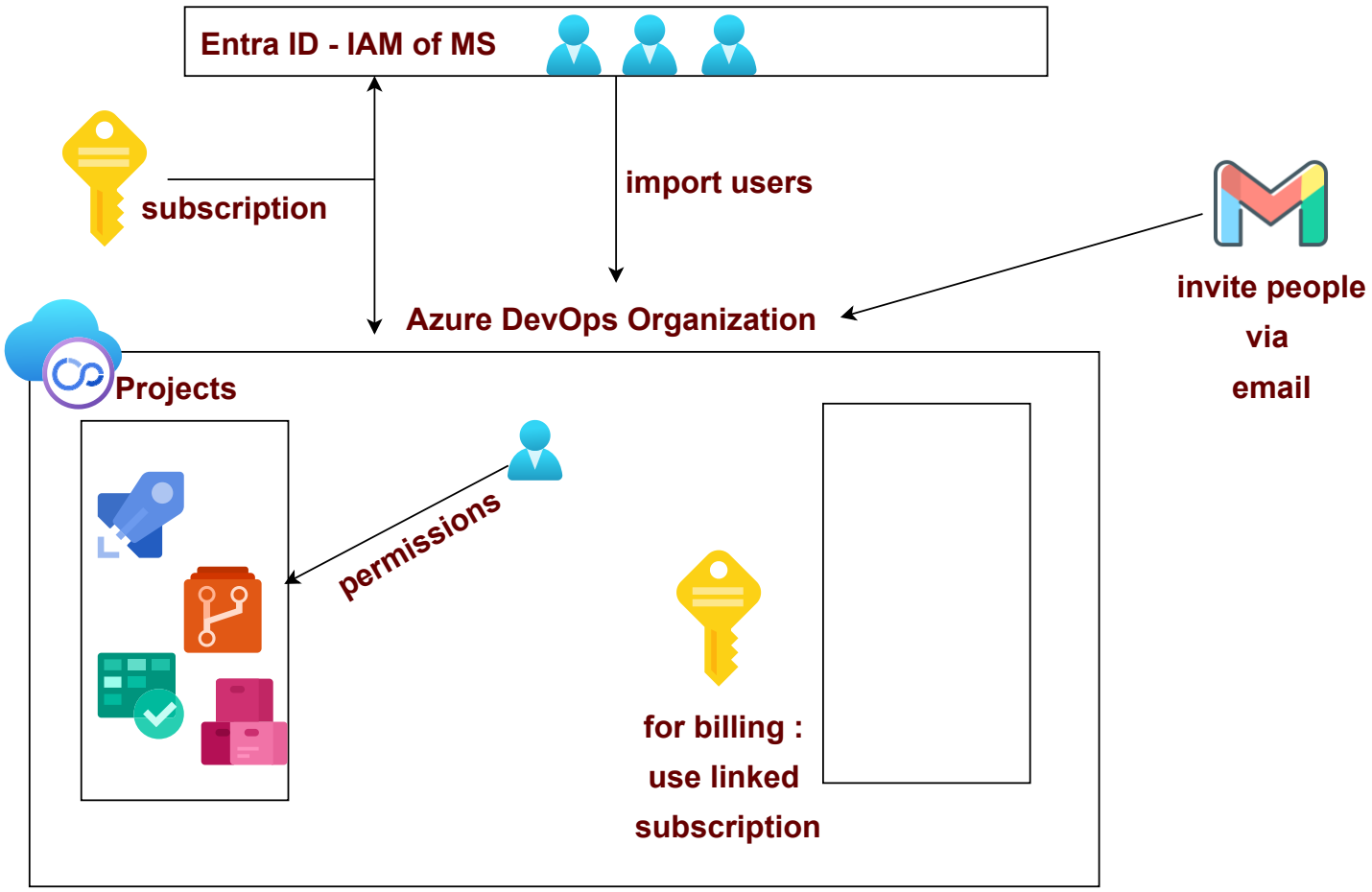
DevOps Methodology uses Tools
to reduce time-2-market

2 Options to Host Azure DevOps:

on-prem: Get a VM (on-prem)
- install and Manage Azure Devops
- Backup, DR, Scaling etc => You
- regulated industry

→ Devops server (enterprises)

→ Devops express (free for learning purposes)



To see services = Access Level

a. Basic: See everything

b. Stakeholder: BOards

c. VS Subscriber : Pay for your membership yourself

Groups

- organization level
- project level

to give

Add Users to Group =

Project Admin, reader etc

Azure BOards Process:

Defines the building blocks of the work item tracking system and supports the Inheritance process model for Azure Boards. This model supports customization of projects through a What You See Is What You Get (WYSIWYG) user interface.

Work Item = Create in Azure Boards to Plan and track your Project

Agile Work Items :

EPIC : Project / Deliverable : Loan Calculator

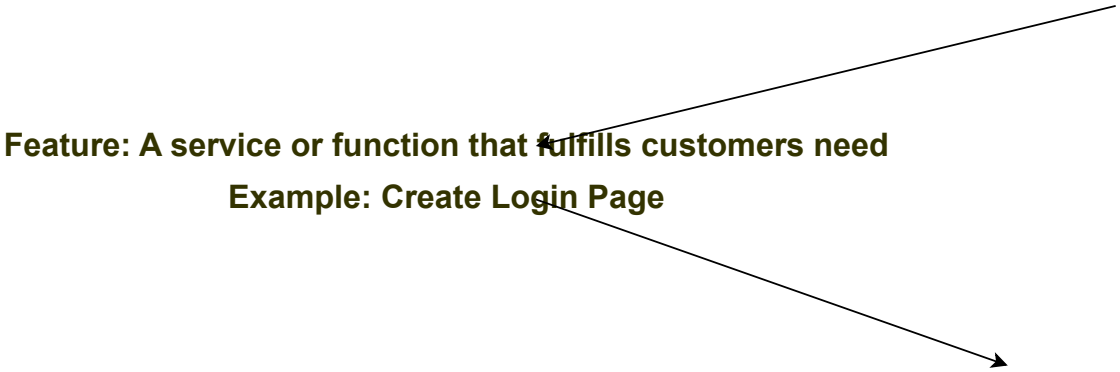
- User Story-1
  - Feature-1
    - Task-1
    - Task-2
    - Task-3
  - Feature-2
- User Story-2

User Story = From End users Perpective how it should look like

As a [Persona], I [want to], [so that]

Example: As A user, I need signin with Google /Facebook Option so that I can sign in quickly

AIM: TO PROVIDE INFO ABOUT HOW THIS BENEFITS USERS



A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). It can help both agile and DevOps teams establish order in their daily work.

If drag feature to last column it disappers from work item

area in azure boards = collection of sprints

users stories = version 1.9.0

user = version 1.10.7

Team = Collection of human beings

- Each Team has there own kanban board

Courier Tracking Website

Frontend (Homepage) + Backend (API + DB)

Person - A,B,c - Frontned

Person-1,2,3 - Backend

TEAM Frontend = Person-A,B,C

TEAM BACKEND = Person-1,2,3

Area Paths - Frontend , Backend

PM -

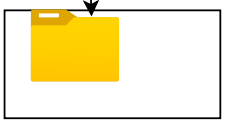
Open boards for Each Area path

Product Backlog = How Frontend + Backend (App)

- user stories of both Frontend, Backend
- How your app works

Teams will pull users stories from
product backlog into THEIR SPRINT BACKLOG

Frontend-Team



Area Path = COllection of
Work Items
for a Team

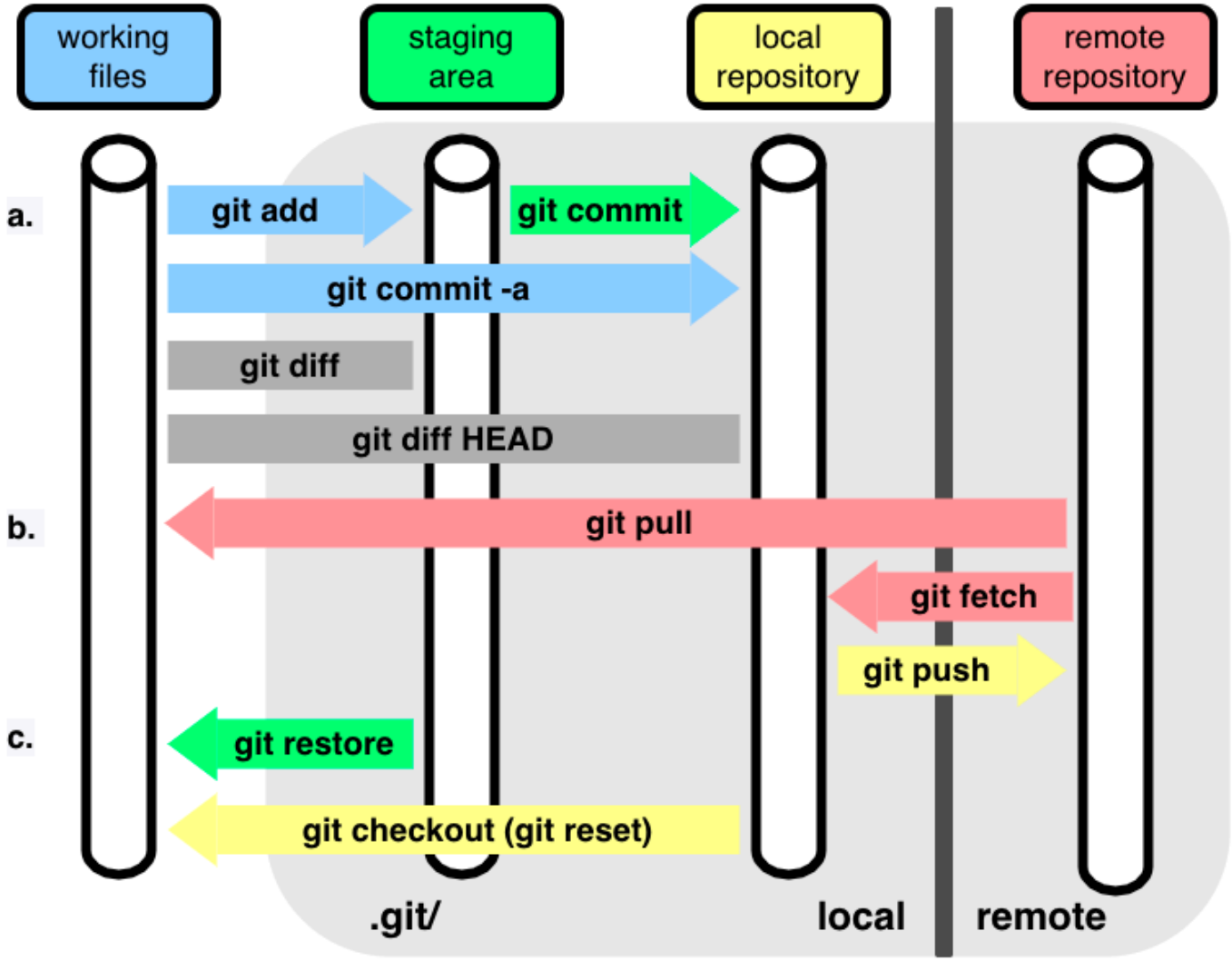
-- It specifies what this team has to do

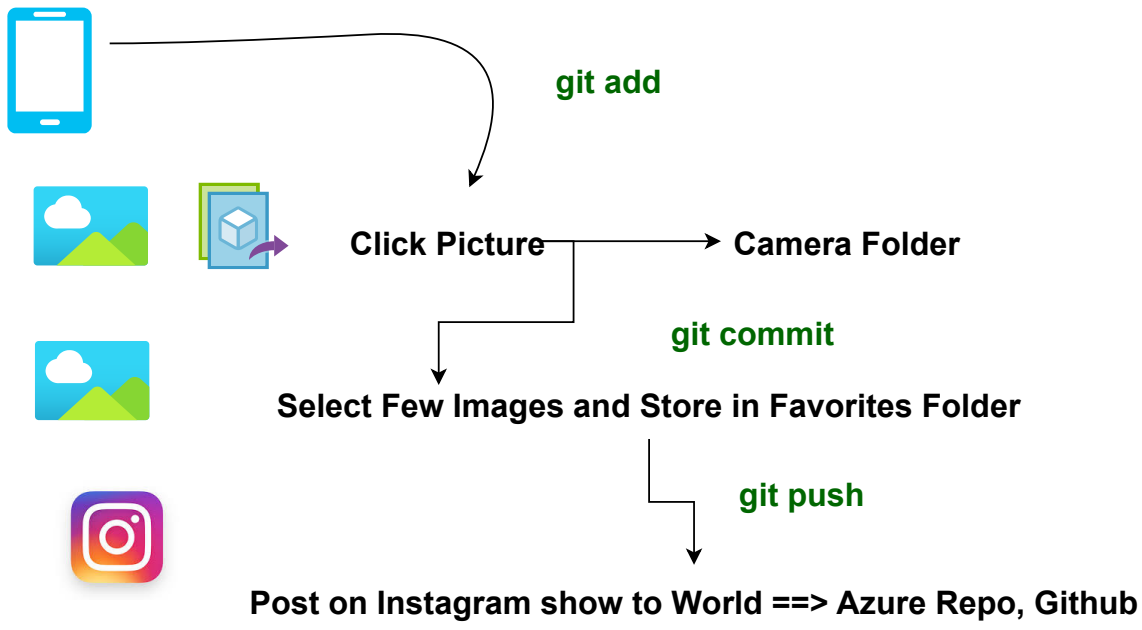
Backlogs: List of work items to perform
according to priority

High level overview
(epic, features, user story)

-- Product Backlog

sprint
what user story to
finish
this week


working files
staging area
local repository
remote repository
a.
git add
git commit
git commit -a
git diff
git diff HEAD
b.
git pull
git fetch
git push
c.
git restore
git checkout (git reset)
.git/
local
remote


git add
Click Picture
Camera Folder
git commit
Select Few Images and Store in Favorites Folder
git push
Post on Instagram show to World ==> Azure Repo, Github

git = software

azure repo, gitlab, github = based on git

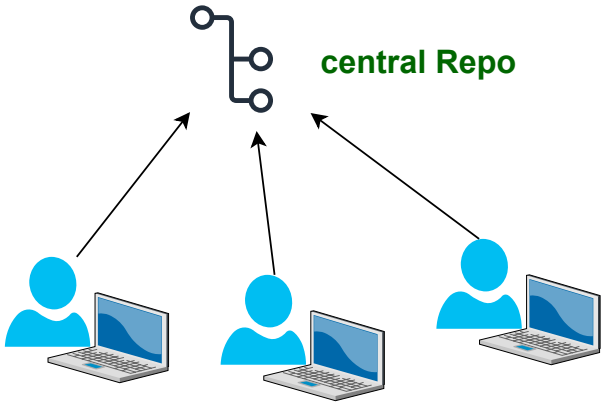
SCM

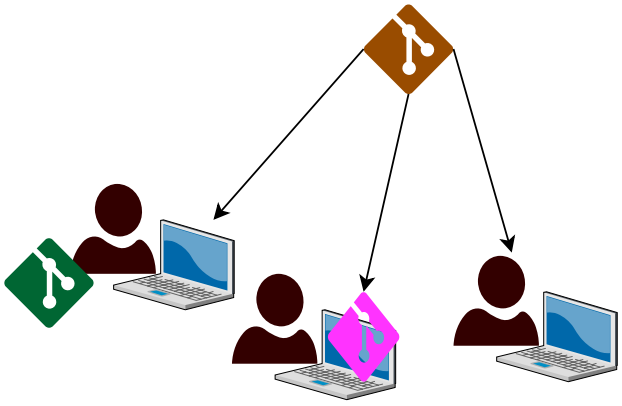
Centralised VCS

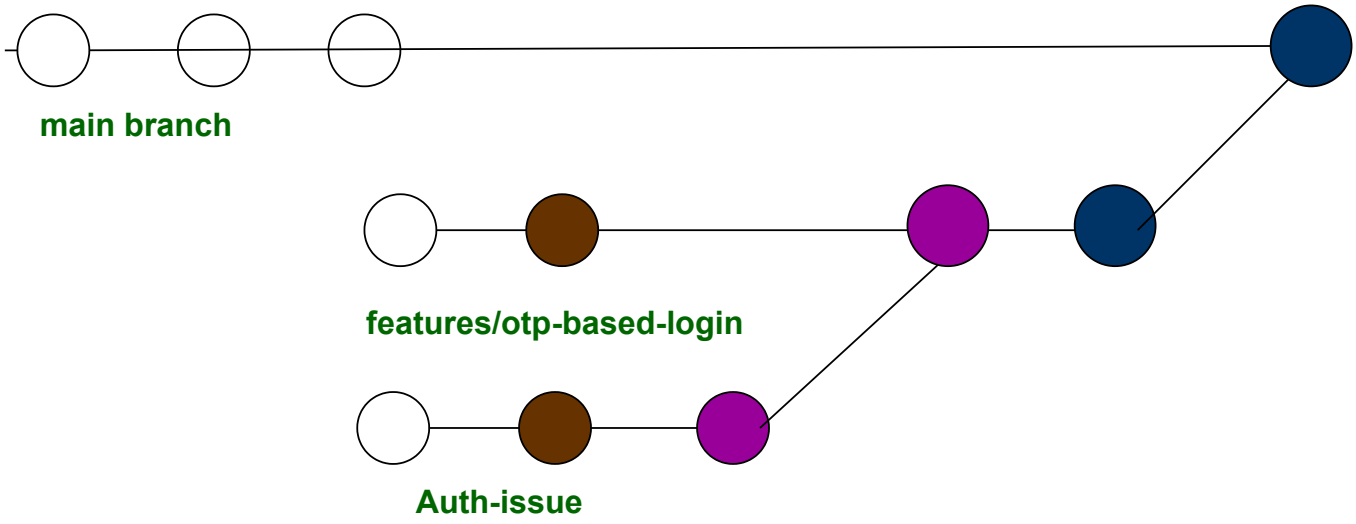
Example - TFVC, Subversion

Distributed

Exmaple - Git, Mercurial







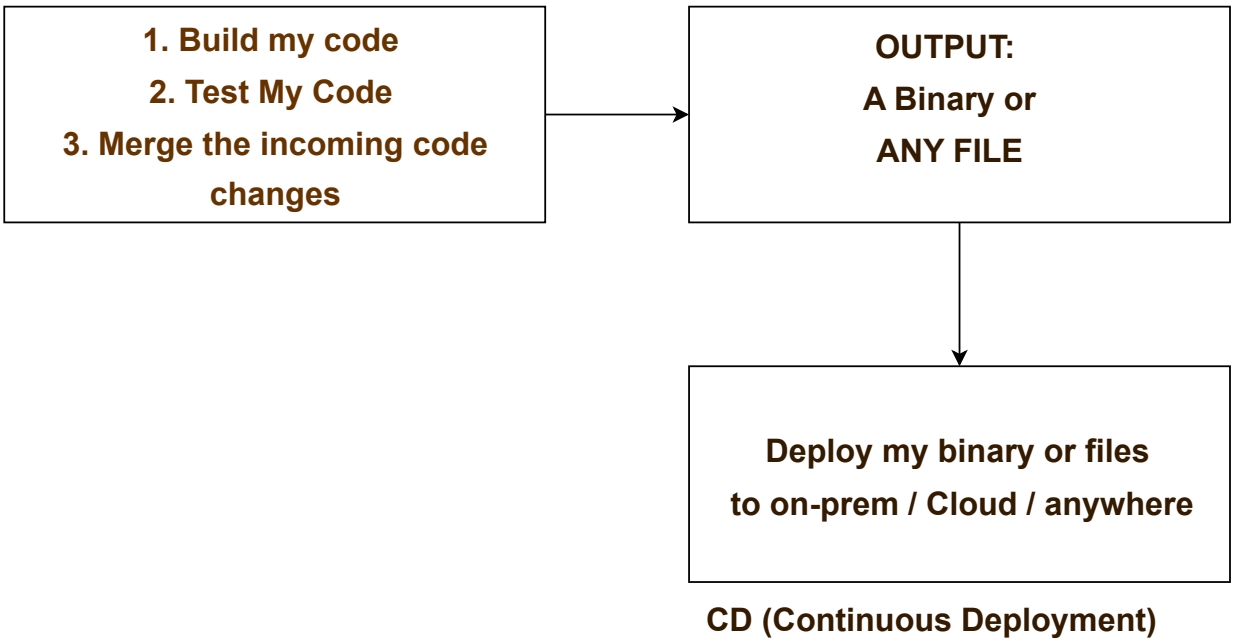
Pull Request is a Mechanism to notify maintainers
that you have completed a task of coding

- Now you wish to merge your branch

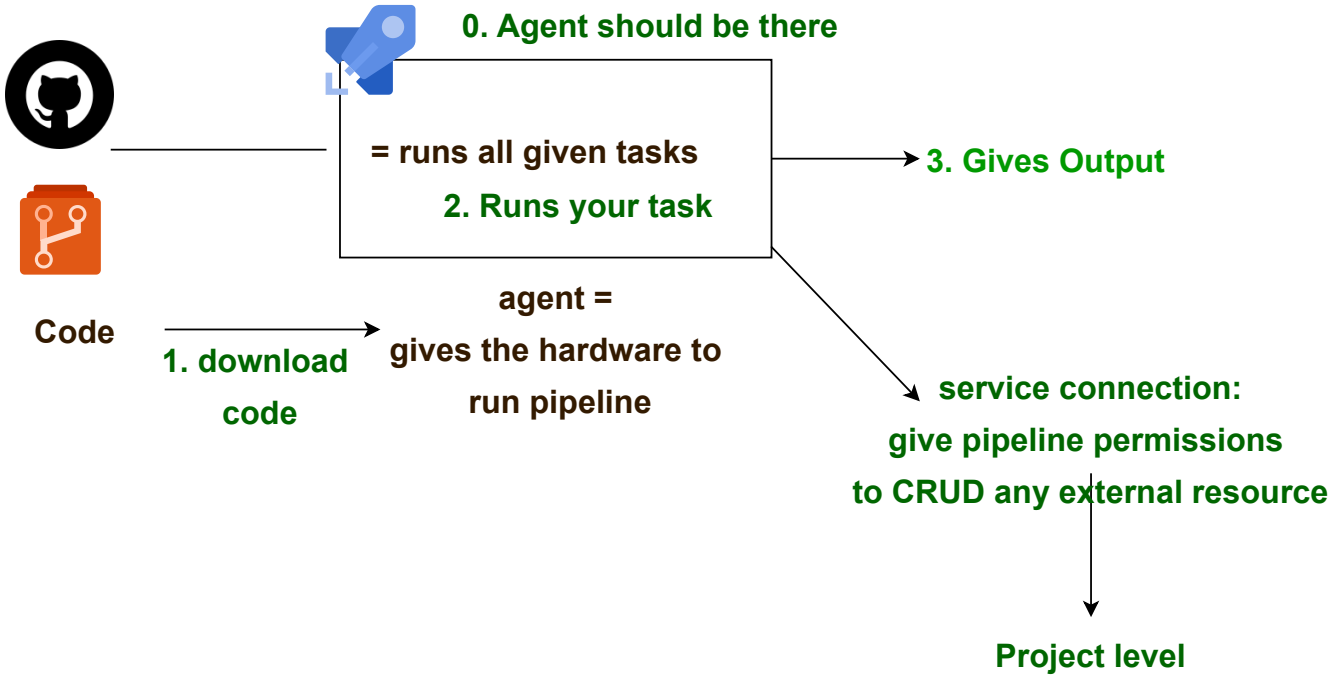
Pipeline: A collection of tasks that runs automatically based
on a trigger

Build Process => Process of converting human readable code
into machine readable binaries (like exe)

A pipeline → **CI (Continuous Integration Pipeline)**



CD (Continuous Deployment)



Agent :

1. Default (MS hosted):
- customers will just use it
- backup, scaling, pre-installed softwares => MS

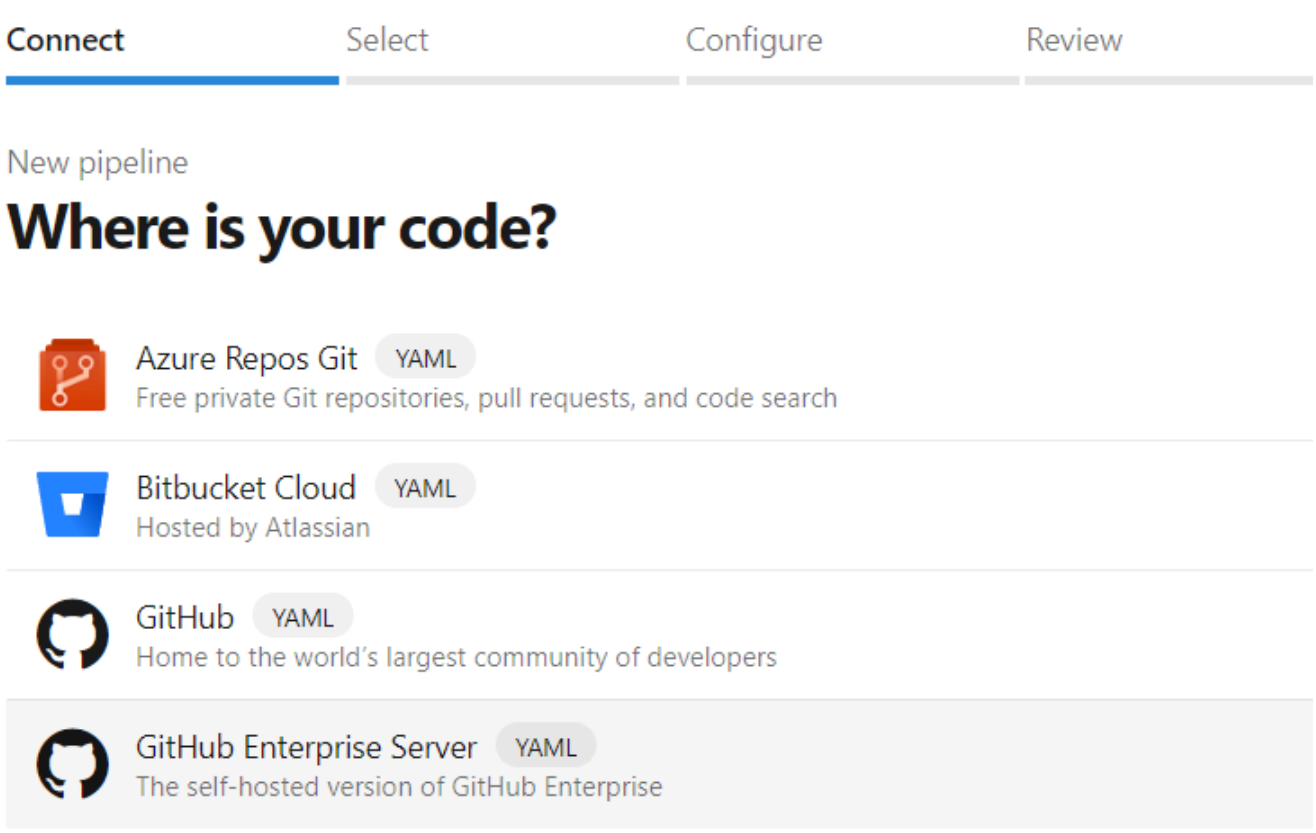
Self Hosted Agent
- customers => upgrade, scaling etc
- you have full control
- example : you need power platform CLI

2 types of pipelines =

1. Classic : GUI
- drag drop get it done
-

2.[Recommended] YAML (pipelines as a code)
- Code in YAML
- keep in Repo (SCM)
- import and share with others

By default only YAML no classic



1 Parallel jobs purchased

1 parallel jobs = 1 agent