**Azure DevOps**

Azure DevOps Server is a Microsoft product that provides version control, reporting, requirements management, project management, automated builds, testing and release management capabilities. It covers the entire application lifecycle and enables DevOps capabilities.

**Software Development Life Cycle(SDLC):**

Previously before DevOps come into picture we used to have SDLC system where the deployment of the application from development environment to production environment done differently these were the steps or stages that we used to have before DevOps:

Code  
 Build **Development Team**  
 Test

Deploy  
 Operate **Operations Team**  
 Monitor

Previously there were separate teams to do the tasks that mentioned above so as DevOps come into picture they combined those tasks and they have made into one team as DevOps team who will do all the works or tasks that mentioned above.

**Benefits of DevOps:**

1. More Agility (Every commit is treated as a final delivery)
2. Easy to use
3. No maintenance
4. Reliable
5. Security

Azure DevOps was formerly known as Visual Studio Team Services (VSTS)

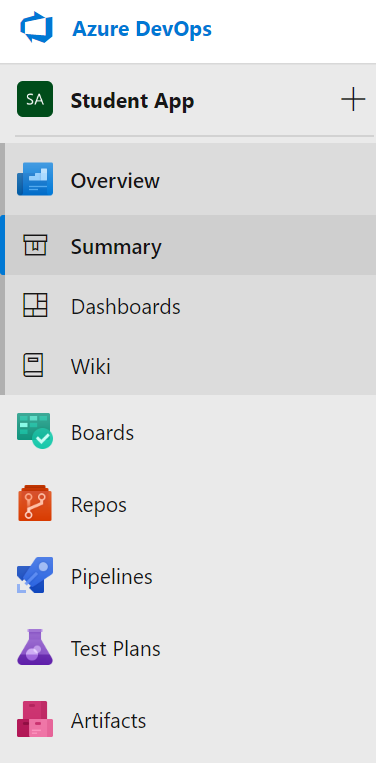
Azure is cloud solution which provides lots of services and features to help your organization in almost all ways during development of any product by providing multiple tools and technologies.

Azure DevOps is a feature or service of Azure.

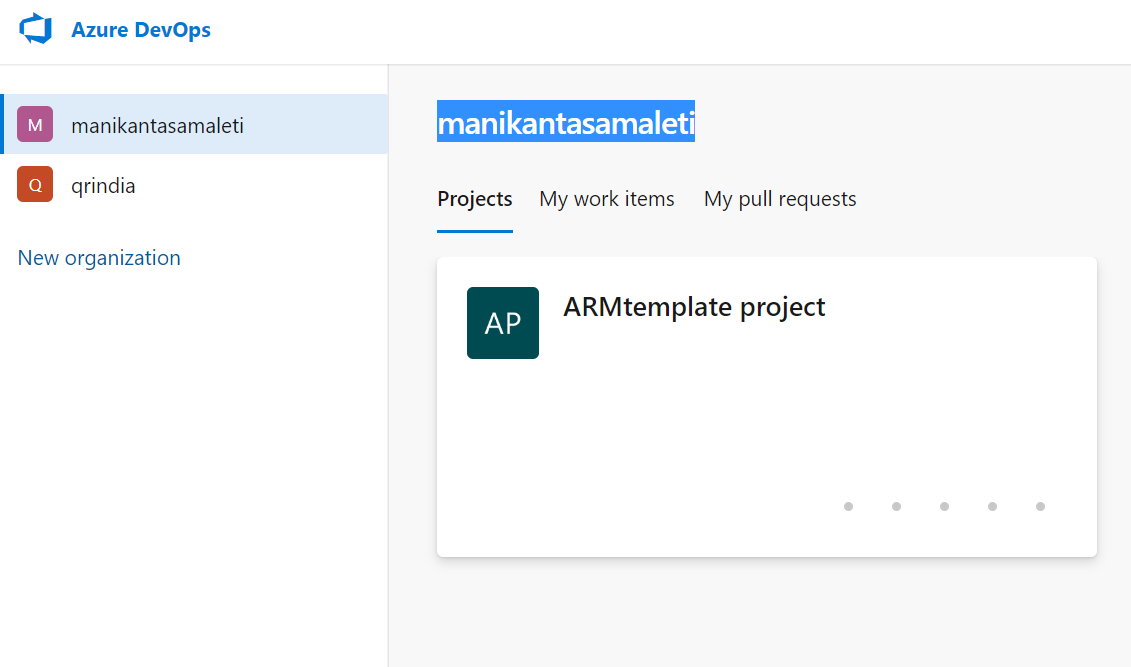
**Azure DevOps Services:**

There are 5 types of services provided by azure devops they are: -

1. Boards
2. Repos
3. Pipelines
4. Test Plans
5. Artifacts



In Azure DevOps when we signup at that time automatically one organization will be created with the name of our email id that we have signed up with.



We can create multiple organizations and in one organization we can create multiple projects.

A project provides a repository for the source code and a place for a group of people to plan, track, progress and collaborate on building software solutions.

All the data related to a software is stored under a project. Each project must have an organization.

There are two different types of projects that we can create in Azdevops those are: -

1. Public project
2. Private project

**Public project: -**

Visible to everyone who has access to internet.

No login required if we have the URL of the project that’s enough to access but we cannot modify the there will be only read-only access.

Each public project has a unique URL.

Public projects are used for open source project development.

We can create unlimited no of projects under one organization.

**Private project: -**

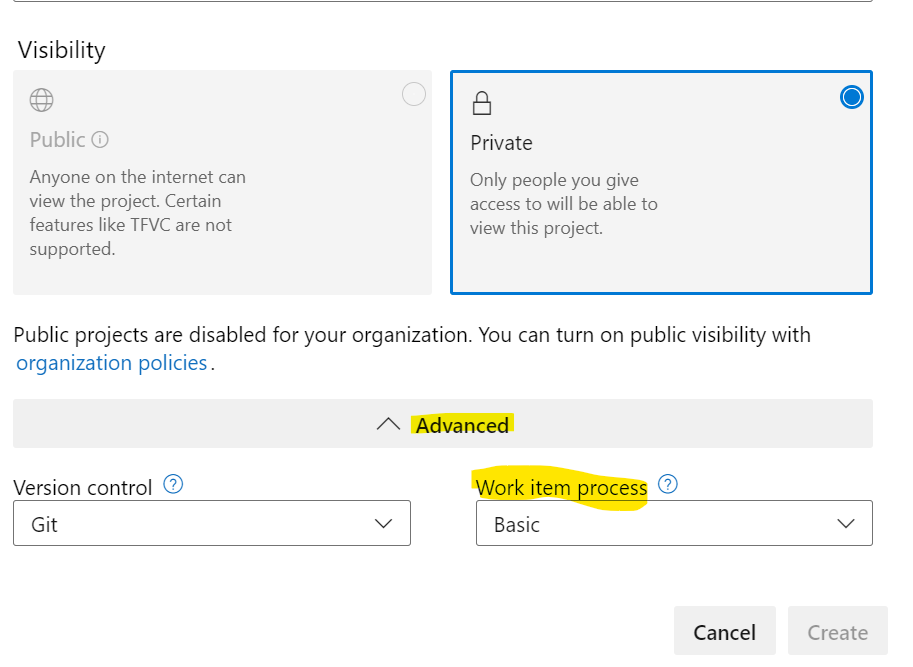
Visible to limited users.

User must login into Azdevops to access the project.

Each private project has a unique URL

While creating the project we will be having an advanced option which is “Work Item Process” in which there will be options of the methodology that you are using the options such as:

1. Basic
2. Agile
3. Scrum
4. CMMI



Graphical user interface, application

Description automatically generated

Note: A user must have administrator level access to make changes in team.

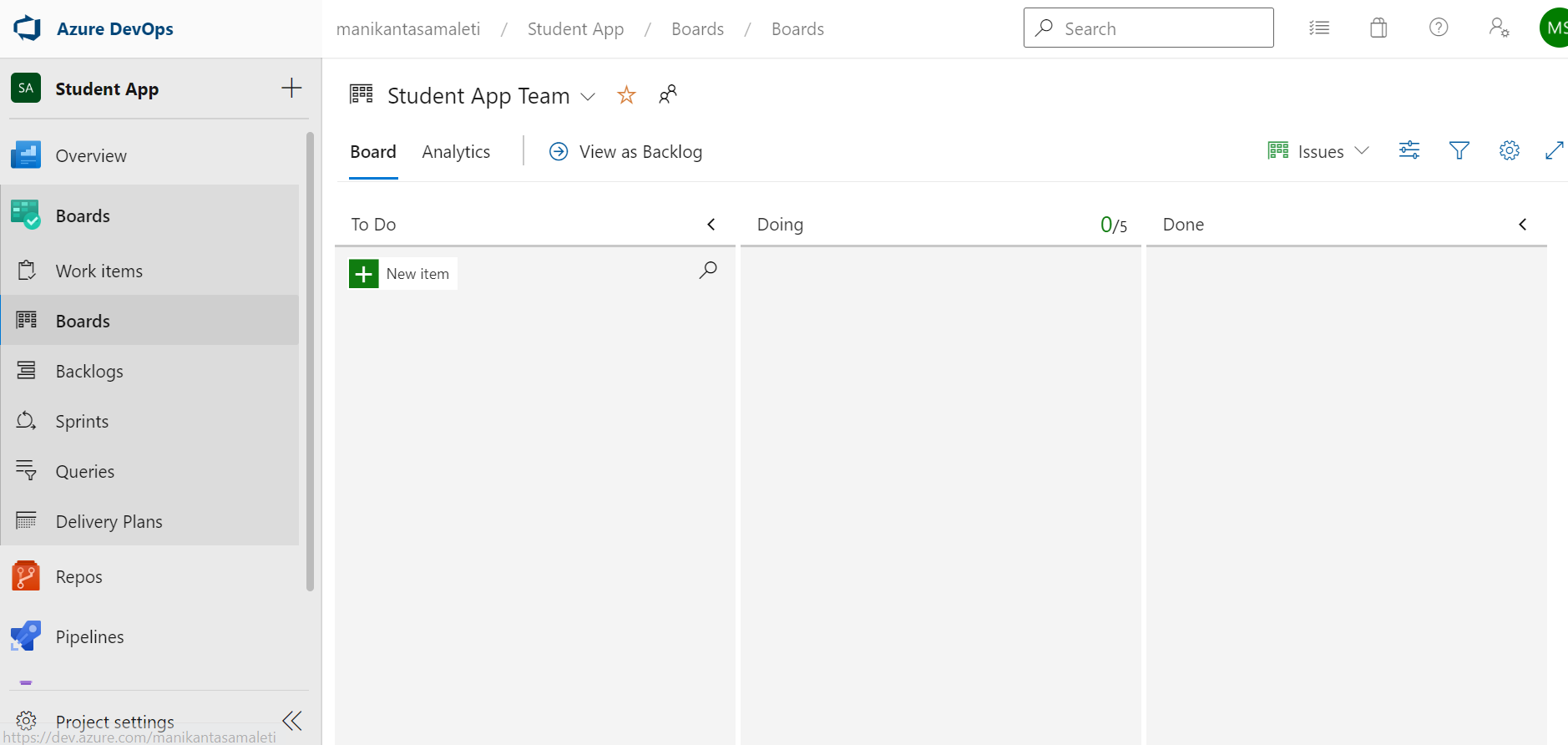
**Azure Boards: -**

Azure Board is the main service of Azdevops used in project to:

1. Track the work with Kanban board.
2. Work with backlog
3. Team dashboard
4. Reporting

A Kanban board is the place where we have all the details of the work items which you deliver or work.

**Kanban Board: -**



A work item is type of work that you are giving or delivering in a project is call work item Example: Bug, ethic, improvement

**Azure Boards are useful in:** -

Tracking the work and progress of work.

You can give access of board to your stakeholder allow him to see the progress.

You can manage scrum, sprint with azure boards.

You can manage product backlog.

**Who will work on Azure boards:** -

The **team**

PO(**Product Owner**) will work to manage the backlog

**Stakeholder** will work to see the status of what team is doing

**Scrum master** will work on board to get different types of reports and to see the progress of team.

The type of the board will be depend on “Work Item process” (Basic,Agile,CMMI,Scrum)

Each type provide different types of “**work item**” and “**work flow**”

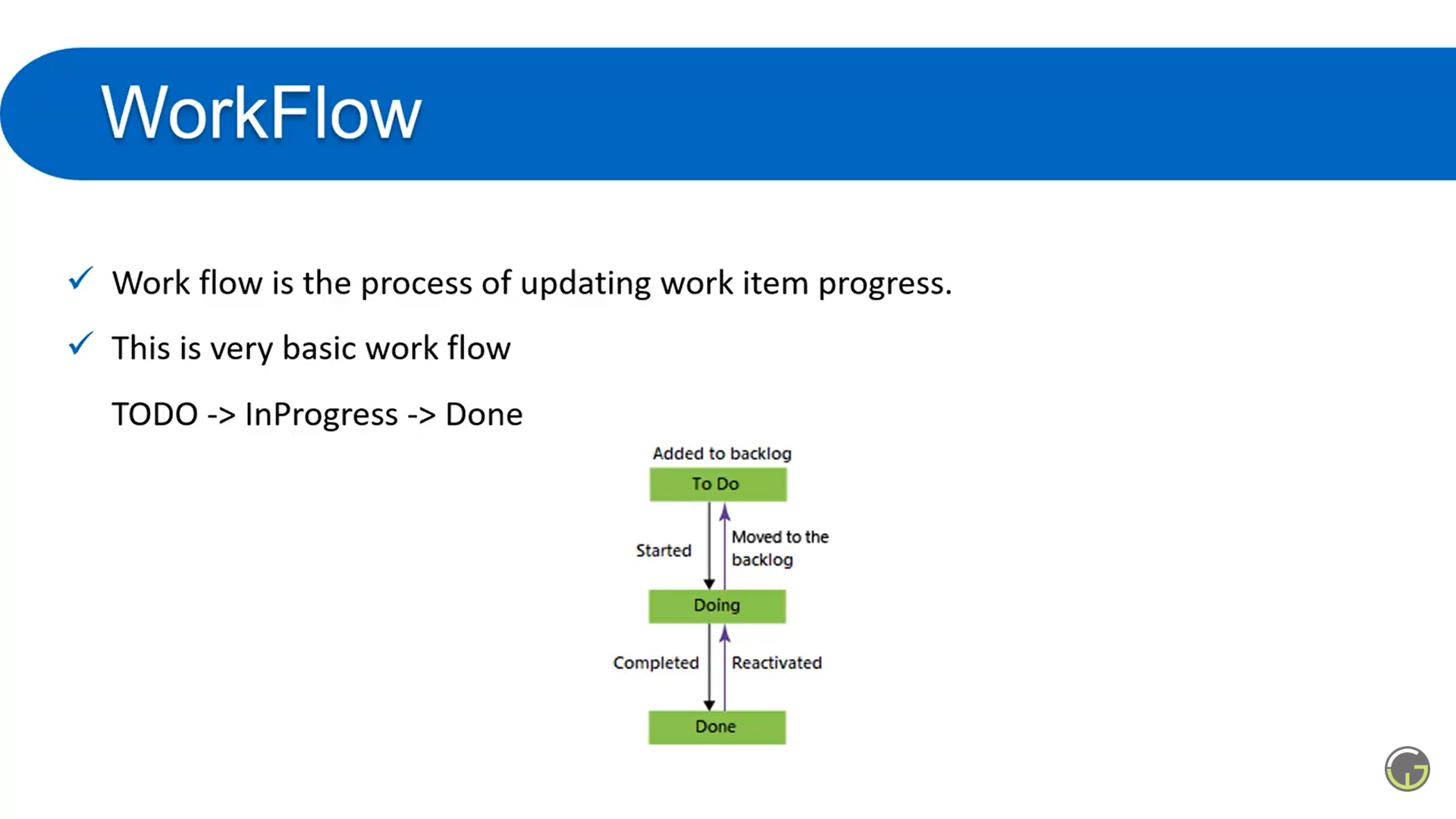
**Work Item:** A work item is a unit (small or large) of work which has several characteristics and is a part of your development.

**Characteristics:**

1. Title
2. Description
3. Assigned to
4. Completed By

**Work Flow:** Work flow is the process of updating the work item progress.

The basic flow is ToDo => Inprogress => Done



**Azure Board with Basic process: -**

There are **3 work items** in azure board with basic process:

1. Epic
2. Issue
3. Task

Diagram

Description automatically generated

Each Epic can have multiple issues and each issue can have multiple tasks

Epic => membership related

Issue => Bug, improvement or user story

Task => It is a tiny part we used to create solve issue

**Backlog:**

A backlog is a collection of work items which will be used for future development.

There are 2 types of backlogs

1. Product Backlog
2. Sprint Backlog

Product backlog is an ordered list of everything that is known to be needed in the product.

Sprint backlog is the collection of work items which are in ToDo list

A sprint is the set period of time during which specific work has to be done and made ready for review

Graphical user interface, text, application, email

Description automatically generated

**Azure Board using Agile process: -**

There are **7 work items** in azure board with agile process:

1. Bug
2. Epic
3. Feature
4. Issue
5. Task
6. Testcase
7. User story

Chart, waterfall chart

Description automatically generated

**Bug**: - Something which is missed or implemented in the wrong way.

**Epic**: - An epic represents a business initiative to be accomplished.

**Feature**: - A feature typically represents a shippable component of software.

**Issue**: - Any other custom type.

**Task**: - Small unit of work.

**Testcase**: - test case for feature.

**User story**: - Implementation of new work.

Diagram

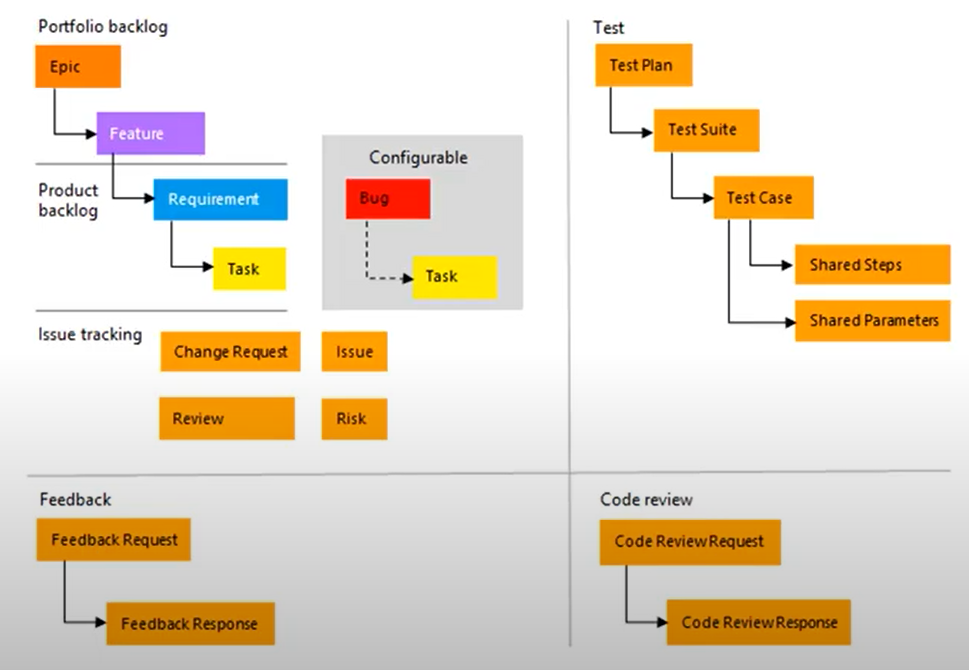
Description automatically generated

**Azure Board using CMMI process: -**

The full form of CMMI is: Capability Maturity Model Integration.

There are **10 work items** present in azure board with CMMI process:

1. Bug
2. Change request
3. Epic
4. Feature
5. Issue
6. Requirement
7. Review
8. Risk
9. Task
10. Test case



**Change request**: If the work done is valid but needs to implement something else

**Requirement**: Work item to track on board.

**Review**: To review some work.

**Risk**: This work item has some risk to implement.

Diagram

Description automatically generated

**Azure Board using scrum process: -**

There are **7 work items** present in the scrum process:

1. Bug
2. Epic
3. Feature
4. Impediment
5. Product backlog item
6. Task
7. Test case

**Chart, waterfall chart

Description automatically generated**

**Impediment:** Depending on the other developer work i,e if other developers work is not completed you cannot continue the work you have.

**Product backlog item**: Work item which will be used to track on board.

Here in scrum process, we call user story as “product backlog item”

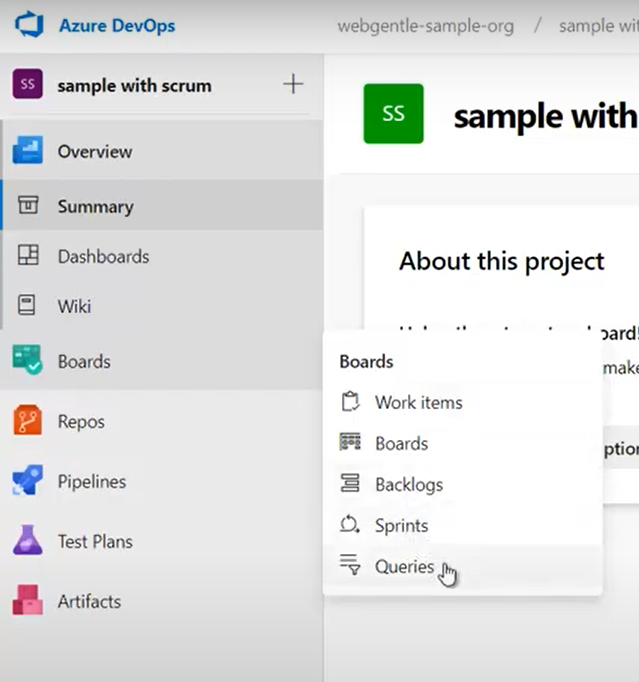
Diagram

Description automatically generated

**Query in Azdevops: -**

**What is Query:** - A query is a combination of few logics which are applied on work items.

**Why do we need Query:** -A query is used to filter work items based on several logic.



Graphical user interface, application

Description automatically generated

**Customize the project using Inherit process: -**

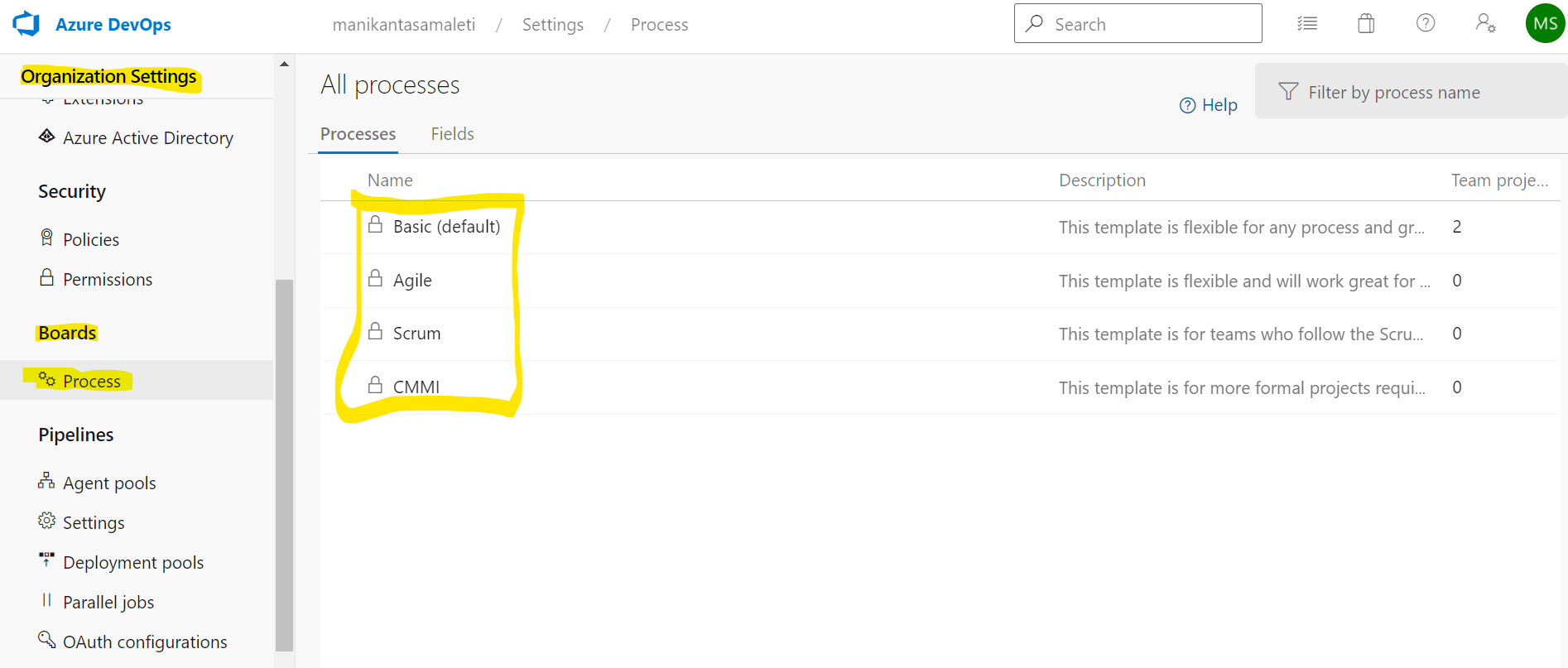
In Azdevops there will be default work processes which we call existing work process they known as: Basic,Agil,CMMI and scrum

So, we cannot customize the existing process if we want to create a customized work process we can use inherited process to create our customized work process.

Inherited process is the extended version of any existing process. Inherited process can be handled at organization level.

Any change you make to the inherited process automatically appears in all the projects that use that process.

You can do this by going into organization settings -> Boards -> Process



**Azure Repos: -**

A repo(Repository) is a container which is used to store and manage your code in a systematic way. In other words repo is a version control tool used to manage code.

Version control systems are type of software which are used to track and manage each and every change in code done by an individual or a team.

In azure repo there are 2 types of version controls are available:

1. Git (Distributed version control)
2. Team foundation version control (centralized version control)

We need to install git in our system for that we can google for “git” and download it install it after installing we should check it whether it is installed or not by going into command prompt (cmd)

**Git commands regarding creating a project in DevOps and pushing the code, pulling the code, creating the branch and switching the branches in the DevOps using Git commands**:

Here there are git commands in orderly manner in creating a project in the DevOps:

First we need to install the Git software in our system to start this the link for the git software is: https://git-scm.com/downloads

After successfully downloading the git software we need to open the command prompt where we can check whether it was installed successfully or not to check that we will use these commands:

**1. git --version** (to check which version of git you installed)

**2. git --help** (to get the list of what all are the git commands we can use in command prompt)

So now we should create one folder in our system and now open that folder using command prompt we can clone our repository that we created in azure DevOps into this folder that we created in our system so to do that we can use one command:

From here onwards whatever we do we need to do in this directory only to make changes happen in this directory:

**3. git clone <here comes the URL of the repo which is present in azure DevOps portal>**

Now we need to create one project using any of the programming languages existing in the world so to do that we have one command:

**4. dotnet new <here comes the template which you want to use to create your project in .net Eg: mvc, APi etc...>**

Now the project has been created and we need to have all the files present in the prject available in the folder so for that we have an command that is:

**5. git add --all**

now we need to do some changes(commits) in our files that we created in our project to have some progress in our project so to update all the changes or commits that we made in our project we will use this command:

**6. git commit -m "Here comes the message you should write here what changes you have made in your project"**

Upto here what we did everything will update in the local repo so now we need to push our commited changes into the local repository.

So for doing that we will use this command:

**7. git push**

At this stage there will be one default barnch which is created when you push your code into the repo in Azure DevOps that default branch we will call it as: master

So if the second developer does not wants to create his own branch and wanted to do changes in his computer then he can pull the code from the server to his computer using this command:

**Note**: for this also we need to select the particular directory to pull the project to your computer:

**8. git pull**

So here if the second developer wants to create his own branch he can go for this command:

**9. git branch <here give the new branch name>**

So once the branch is created if he wants to switch his branch from master to his own branch he can use this command:

**10. git checkout <his branch name>**

To know which branch at you are you can use this command:

**11. git branch**

Now after all the commits that person made and finally if he wants to push his code into Azure DevOps server he can sue this command:

**12. git push --set -upstream origin <here comes branch name (of server)>**

**If the project is already created in visual studio** and now you want that project to be there in your local repository so at this stage you can do this

First open the folder where you have created your project in command prompt then you can use this command to initialize the "git" as a version control system for that project for that we will use this command:

**1. git init**

To push the existing repository from command line we can use these commands:

**2. git remote add origin <Here comes the project URL from server>** (it will be available in the Azure DevOps server you can simply copy and paste the commands in command prompt along with project URL)

**What is Branch in repo:**

The branch is the place or thing where we can keep track of all the commits done by any person or developer.

There will be a default branch and the main branch that is called as “master”

**Why do we need branches:**

Main purpose is to avoid breaking change

To develop new feature (Eg: Authentication)

POC (Proof of concept)

People create branch per story development

We can create branch per developer

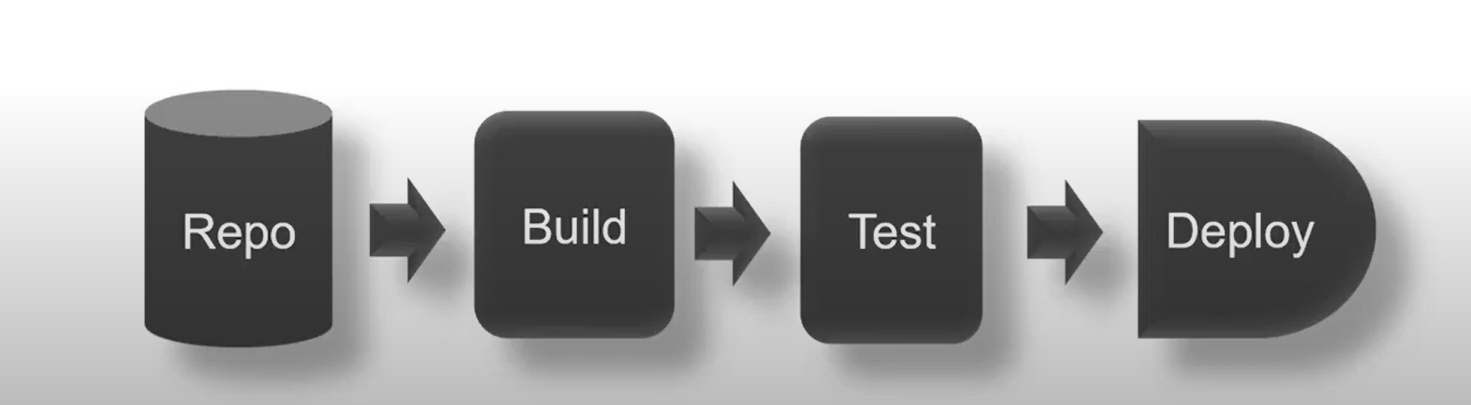
**Azure Pipelines: -**

Graphical user interface, text, application

Description automatically generated

Pipeline in DevOps is a set of process (Automated or Manually triggered) which is used to make available your project code to users.

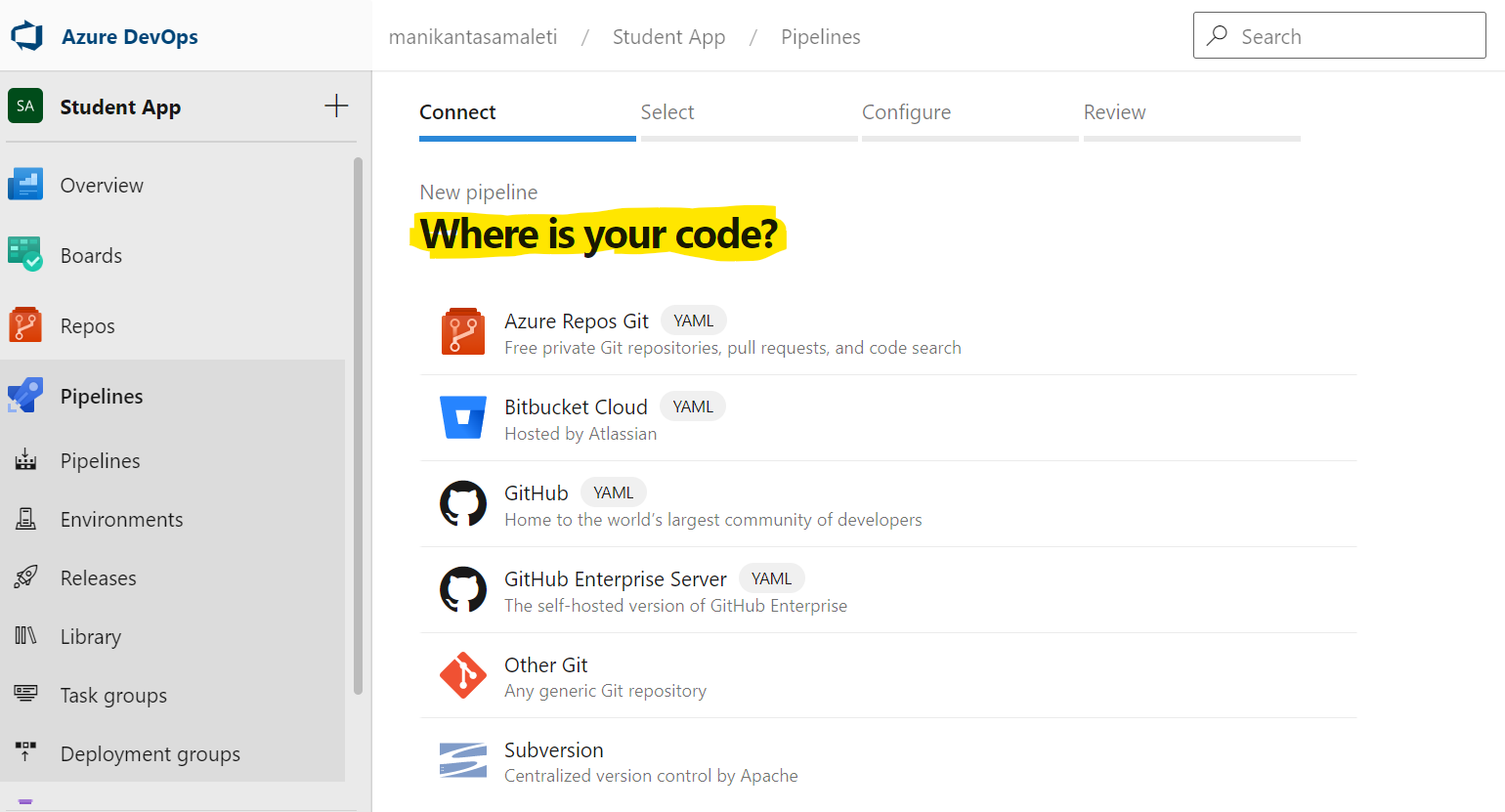
Azure pipelines is a cloud service that you can use to automatically build and test your code project and make it available to other users.



We will follow these steps to deploy our project code and make it available for the users so here in this process if our project doesn’t need any testing we can remove that testing stage by using this pipeline process.

Azdevops works with any programming language and any platform  
Eg: .net, c#, python, ruby, php, Java.

**Source code:**

****

Your project source code (repo) must be in a version control system

Azure pipelines integrates with:

Github  
Github Enterprise  
Azure repos git and tfvc  
Bitbucket cloud  
Subversion

If your code is available in any of the written repositories then you can use Azdevops pipelines to build and deploy your code.

**Deployment Target: -**

It is a location to deploy the code. Azdevops can be used to deploy your code on multiple targets

Eg: Containers, Vm's, Azure service etc...

