**Assignment-1**

1. Write the steps “Using Git how to push the file in Bitbucket / Gitlab / Git hub “

**Bitbucket:**

• Create a new repository in the bitbucket

• Means creating a new repo with template

• Then creating a new folder same in the project as given in the repository

• And adding text content files or text format files with content adding into new folder

• The folder is sharing to the Git by share

• First command is **git init**

• It shows reintialized existing Git repository in c:/Users/Lekhana/OneDrive/Desktop/row/.git

• Next command is **git status** used show how many files in the folder

• Next command is **ls** also used to show how many files in the folder

• Next command is **git config** is a convenience function that is used to set Git configuration values on a global or local project level

• Command is **git config --global user. email** “lekhana.neethipudi20@gmail.com”[type the email is the login in the bitbucket ]

• Then **git config –global user.name** “lekhana\_Neethipudi” [ type the name is given in login name of bitbucket]

• Command is git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols . The git clone command copies an existing git repository.

• Copy the text from the clone in https and paste it in git bash and enter

• Command is **git clone https://lekhana\_neethipudi@bitbucket.org/lekhana-neethipudi/rowrepo.git**

• Now it shows a dialogue box enter email, password and type the email is connected to bitbucket and type password correctly then authentication is verified.

• Then again go to git bash type command **git add** . is used to add the files in to the repository.

• And type command git **add text.txt** then the files of the folder

• Then next command is git commit is one of the core primary function of git. Prior use of the git add command is required to select the changes that will be staged for the next commit.

• **git commit -m “first commit”**

• Next command **git add repo** is adding the repo to the git bash

• Then the git show command is used show the files in the folder

• The next command is **git remote add origin master https URL** [ The URL is in the repository of clone copy and paste it]

• The command is **git push -f origin master**

• Go to the bitbucket reload the repository and it shows the extra file is add in the existing files.

* https://bitbucket.org/lekhana-neethipudi/rowrepo/src/master/

**Git lab:**

• Create a new project/repository in Gitlab

• Means creating a new project

• Then creating a new folder same in the project as given in the project

• And adding text content files or text format files with content adding into new folder

• The folder is sharing to the Git by share

• First command is **git init**

• It shows reintialized existing Git repository in c:/Users/Lekhana/OneDrive/Desktop/horizontal/.git

• Next command is **git status** used show how many files in the folder

• Next command is **ls** also used to show how many files in the folder

• Next command is **git config** is a convenience function that is used to set Git configuration values on a global or local project level

• Command is **git config --global user. email** “lekhana.neethipudi20@gmail.com”[type the email is the login in the git lab]

• Then **git config –global user.name** “lekhana\_Neethipudi” [ type the name is given in login name of git lab]

• Command is git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols . The git clone command copies an existing of clone https URL repository/project in the git lab

• Copy the text from the clone in https and paste it in git bash and enter

• Command is **git clone** [**https://git**](https://git) **lab/lekhana\_neethipudi/horizontal. git**

• Now it shows a dialogue box enter email, password and type the email is connected to git lab and type password correctly then authentication is verified.

• Then again go to git bash type command **git add** . is used to add the files in to the repository/project.

• And type command git **add text.txt** then the files of the folder

• Then next command is git commit is one of the core primary function of git. Prior use of the git add command is required to select the changes that will be staged for the next commit.

• **git commit -m “first commit”**

• Next command **git add horizontal** is adding the repo to the git bash

• Then the git show command is used show the files in the folder

• The next command is **git remote add origin master https URL** [ The URL is in the repository/project of clone copy and paste it]

• The command is **git push -f origin master**

• Go to the git lab reload the repository and it shows the extra file is add in the existing files.

* In the top of project shows the message like **“ You pushed to master at lekhana\_Neethipudi /horizontal”**
* https://gitlab.com/Lekhana\_Neethipudi/horizontal

**Git hub**

• Create a new project/repository in Git hub

• Means creating a new project

• Then creating a new folder same in the project as given in the project

• And adding text content files or text format files with content adding into new folder

• The folder is sharing to the Git by share

• First command is **git init**

• It shows reintialized existing Git repository in c:/Users/Lekhana/OneDrive/Desktop/vertical/.git

• Next command is **git status** used show how many files in the folder

• Next command is **ls** also used to show how many files in the folder

• Next command is **git config** is a convenience function that is used to set Git configuration values on a global or local project level

• Command is **git config --global user. email** “lekhana.neethipudi20@gmail.com”[type the email is the login in the git hub]

• Then **git config –global user.name** “lekhana\_Neethipudi” [ type the name is given in login name of git hub]

• Command is git code is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols . The git code command copies an existing of code https URL repository/project in the git hub

• Copy the text from the code in https and paste it in git bash and enter

• Command is **git clone** [**https://git**](https://git) **hub/lekhana\_neethipudi/vertical. git**

• Now it shows a dialogue box enter email, password and type the email is connected to git lab and type password correctly then authentication is verified.

• Then again go to git bash type command **git add** . is used to add the files in to the repository/project.

• And type command git **add text.txt** then the files of the folder

• Then next command is git commit is one of the core primary function of git. Prior use of the git add command is required to select the changes that will be staged for the next commit.

• **git commit -m “first commit”**

• Next command **git add horizontal** is adding the repo to the git bash

• Then the git show command is used show the files in the folder

• The next command is **git remote add origin master https URL** [ The URL is in the repository/project of code copy and paste it]

• The command is **git push -f origin master**

• Go to the git hub reload the repository and it shows the extra file is add in the existing files.

* In the top of project shows the message like **“master had recent pushes”**
* https://github.com/lekhana143/vertical

2. Write a program in c to sort the elements of Array (Data structure)?

#include<stdio.h>

int main()

{

int a[6]= {13,8,4,11,5,2};

int temp;

int i, j;

printf("Before Sorting ");

for(i=0; i<6; i++)

{

printf("%d ",a[i]);

}

for(i=0; i<6; i++)

{

for(j=i+1; j<6; j++)

{ if(a[i]>a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

printf("\nAfter Sorting ");

for(i=0; i<6; i++)

{

printf("%d ",a[i]);

}

return 0;

}

Output :

Before sorting 13 8 4 11 5 2

After sorting 2 4 5 8 11 13

3. Explain Water fall model, Agile and Devops in brief

**Waterfall Model:**

The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases.

**Advantages:**

1. Easy to use, simple and understandable

2. Easy to manage as each phase has specific outputs and review process

**Dis Advantages:**

1. It doesn’t allow much reflection or revision. When the product is in testing phase, it is very difficult to go back and change something which is left during the requirement analysis phase.

2. Risk and uncertainty are high.

3. Only forward phases and silos testing.

**Agile:**

* Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.
* The authors of the Agile Manifesto chose “Agile” as the label for this whole idea because that word represented the adaptiveness and response to change which was so important to their approach.
* It’s really about thinking through how you can understand what’s going on in the environment that you’re in today, identify what uncertainty you’re facing, and figure out how you can adapt to that as you go along.
* gile software development is more than frameworks such as Scrum, Extreme Programming, or Feature-Driven Development (FDD).
* Agile software development is more than practices such as pair programming, test-driven development, stand-ups, planning sessions, and sprints.

**Devops:**

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

* the Key Components of DevOps
* Continuous Development.
* Continuous Integration.
* Continuous Testing.
* Continuous Feedback.

Jenkins is an open source continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the Java programming language. It is used to implement CI/CD workflows, called pipelines.

Jenkins is a Java-based open-source automation platform with plugins designed for continuous integration. It is used to continually create and test software projects, making it easier for developers and DevOps engineers to integrate changes to the project and for consumers to get a new build.

4. What is Operating Systems and write 20 commands of linux operating system?

**Operating system**

An operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer. The application programs make use of the operating system by making requests for services through a defined application program interface (API). In addition, users can interact directly with the operating system through a user interface, such as a command-line interface (CLI) or a graphical UI (GUI).

In a multitasking operating system, where multiple programs can be running at the same time, the OS determines which applications should run in what order and how much time should be allowed for each application before giving another application a turn.

It handles input/output (I/O) to and from attached hardware devices, such as hard disks, printers and dial-up ports.

**Linux commands:**

1)**clear command** : clear is a standard operating system command

syntax:$clear

2)**Exit command**: this command is used to close the terminal window or exit the shell

syntax:$ exit[n]

3)**echo command**: The echo command is used to print the line of text or string or a number.

syntax:$echo [option][string]

4)**history command**: The history command displays all previously used commands since the session was started.

Syntax: $ history

5) **pwd command**: pwd stands for present working directory which shows the path of the working directory starting from the root.

Syntax:$ pwd

6) **cd command**: The 'cd' command is used to change the current working directory into its child or parent directory or any other directory (home or root directory).

Syntax: $cd [directory\_name]

7**) ls Command**: The ls command shows the list of all the files and folders in the current working directory.

Syntax: $ ls [option]

8) **mkdir command**: mkdir stands for make directory and it is used to create a new directory in a file system. It takes one or more argument as a directory name. It also creates multiple directories by passing their directory names in the command. By using this command, you can also set permission to the directory.

Syntax: $ mkdir [option] directory\_name

9) **rmdir command**:It stands for remove directory.

Syntax1: $ rmdir [option] directory\_name

10) **cp command**: The 'cp' command is used to copy the file or directory from one location (source) to another location (destination). This command takes at least two arguments as a file name. It can also copy more than one file in a single directory.

Syntax: $ cp [option] Source Destination

11) **mv command**: ‘mv’ stands for move and is used to move one and more files and directories from one place to another directory or location in a Linux file system.

Syntax: $ mv [option] source\_directory destination\_directory

12) **touch command**: The touch command is a standard command used in Linux for creating the empty files if not exits. It is also used to create, change and modify the timestamp (i.e., date and time) of the file. It takes at least one argument as a file name.

Syntax:$ touch file\_name

13) **cat command**: cat (concatenate) command reads the data from the file and displays the content of the file as an output on the terminal. It also helps to append the content of a text file to the end of another text file.Display the content of a file.

Syntax: $ cat [option] file\_name

14) **ln command**: ln command is used to link two files and directories in Linux.

Syntax: $ ln file1 link1

15) **date command**: The ‘date’ command is used to display the current date and time of the system. A super-user can change the value of the system date and time.

Syntax:

$ date [option] [Format]

16) **ps command**: ‘ps’ stands for ‘processes status,’ and this command is used to display the currently running processes with their process Identification number, i.e., PIDs along with some other information for each process depends on the different options.

Syntax:$ ps [option]

17) **whoami command:** whoami is mainly the concatenation of the string “who” ,”am”,”i” and when this command is executed, then it displays the username of the current user.

Syntax:$whoami [option]

8) **passwd command:** The passwd command changes the password of a user account in Linux

Syntax:$passwd

19) **uname command**: The uname command shows the information about the system, such as machine name, Operating system, and kernel.

Syntax:$ uname [option]

20) **apt command:** The Debian based ‘apt’ command stands for Advanced Package Tool. Apt is a package manager for Debian based systems like Ubuntu, Kubuntu, Lubuntu, etc., that automatically and intelligently search, update, install, and resolves dependency of packages on Linux system from command line Interface.

Syntax:$apt-get package\_name

21. Vi/vim command : This is used to edit the existing content

Syntax : $ vi a / $ vim a

22. nano command: this is used to edit the existing content

Syntax : $ nano a

5.What is shell script? Write program for 1. Hello world 2. Variable 3. Operators 4.control statement 5.function

**Shell script**:

A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter. A shell script usually has comments that describe the steps. The different operations performed by shell scripts are program execution, fileA shell script is a text file that contains a sequence of commands for a UNIX-based operating system.

It is called a shell script because it combines a sequence of commands, that would otherwise have to be typed into the keyboard one at a time, into a single script. manipulation and text printing. A wrapper is also a kind of shell script that creates the program environment, runs the program etc.

types of Shell

Bourne Shell. Bourne shell (sh) Korn Shell (ksh) Bourne Again shell (bash) POSIX shell (sh)

C Shell.

steps in creating a shell script

* Create a file using a vi editor(or any other editor). Name script file with extension .sh.
* Start the script with #! /bin/sh.
* Write some code.
* Save the script file as filename.sh.
* For executing the script type bash filename.sh

**1.Hello world:**

#! /bin/bash

# this is program hello world

echo “hello world”

2. **Variables:**

#! /bin/bash

# program variables

Echo “enter a value”

read a

echo “enter b value”

read b

echo “value a” $a

echo “value b” $b

**3. Operators:**

Arithmetic operator:

#! / bin/bash

# program for operators

A=20

B=10

C= `expr $a + $b`

d= `expr $a - $b`

e= `expr $a /\* $b`

f= `expr $a %$b`

echo “sum is “ $c

echo “sub is “ $d

echo “mul is “ $e

echo “div is “ $f

Relational operator:

#! /bin/bash

# program relational

Echo “enter a value”

read a

echo “enter b value”

read b

if [ $a == $b ]

then

echo “a is equal to b”

else

echo “ a not equal to b”

fi

logical operator:

#! /bin/bash

#program logical operator

a = true

b= false

If [$a == true && $ b == true]

then

echo “both are true”

else

echo “both are not true”

fi

**4.Control Statement:**

If else:

#! /bin/bash

# program if else

Echo “enter a value”

read a

echo “enter b value”

read b

if [ $a < $b ]

then

echo “a is greater than b”

else

echo “b is greater than a”

fi

while:

#! /bin/bash

# program while

i=1

while [ $i < 10 ]

do

echo “ $i => lekhana”

i= `expr $i+1`

done

for :

#! /bin/bash/

#program for

for [ i=10 ; i>=1;i--]

do echo "$i"

done

**5.Function:**

#!/ bin/bash

# function program

Echo “enter a”

Read a

Echo “enter b”

Read b

Function add()

{

Echo `expr $a + $b`

}

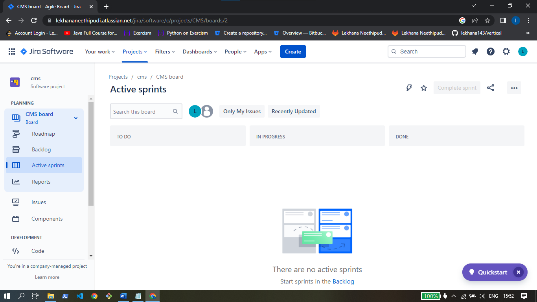
Add

6.Write steps in Jira with screen shot for creating project, epic, story, task, sprints?

Jira Software is the agile project management tool used by teams to plan, track, release and support world-class software with confidence. It is the single source of truth for your entire development lifecycle, empowering autonomous teams with the context to move quickly while staying connected to the greater business goal. Whether used to manage simple projects or to power your DevOps practices, Jira Software makes it easy for teams to move work forward, stay aligned, and communicate in context.

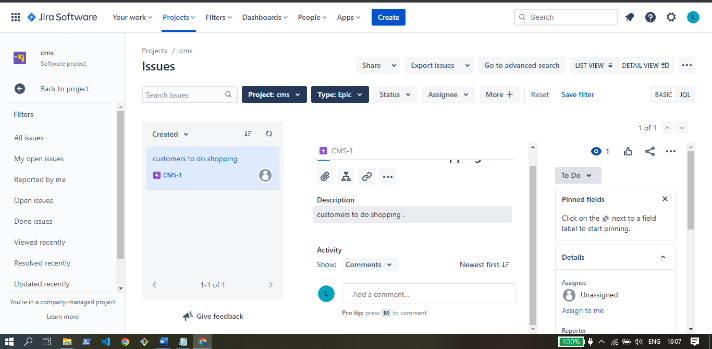
Creating project:

* First creating account in Jira software
* Then go to the menu in Jira page
* It`s contain create option click on create button then its shows a dialogue box
* It`s contains a software development template then click on scrum master or select the scrum master template and again it show a dialogue box now click on use template
* Click on self -managed or company managed
* Then again it shows a dialogue box containing name of the project enter the name of the project and click on create project.



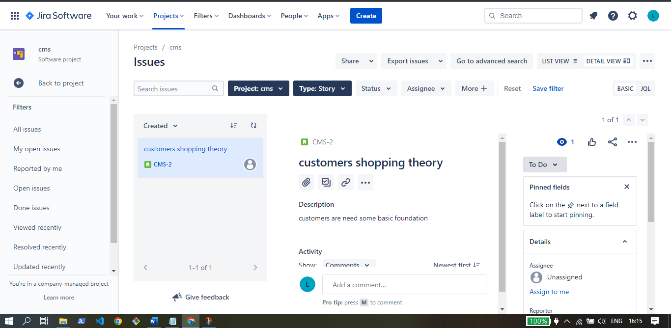
Epic:

* Now opens the project now click the issues on the dashboard of the project
* Now click on create and its shows dialoge box create issue
* The box contains so many options are given below
* the name of the project
* issue type dialogue box
* now select epic
* now type epic name text box
* summary option
* type summary about the issue
* Attachment :attach the file from browse or any file required
* Description option is used type the large description about the project
* and having assign option it helps to assign issue to some one
* and click on create issue



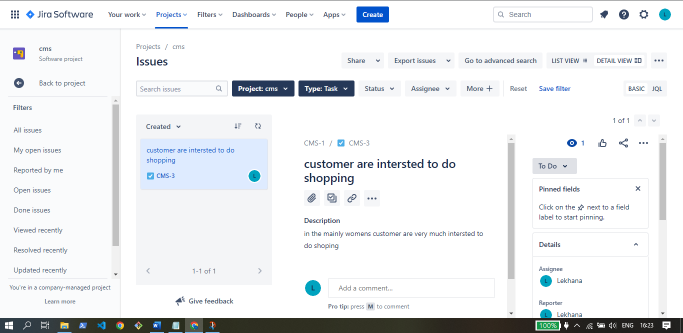
Story:

* Now again click on create issue
* project: the name of the project
* issue type dialogue box and now select story
* summary option is used to type summary about the issue
* Attachment
* attach the file from browse or any file required
* Description is used type the large description about the project
* and having assign option it helps to assign issue to some one
* and epic link option its used connected last epic
* sprint option to select active sprint
* and click on create



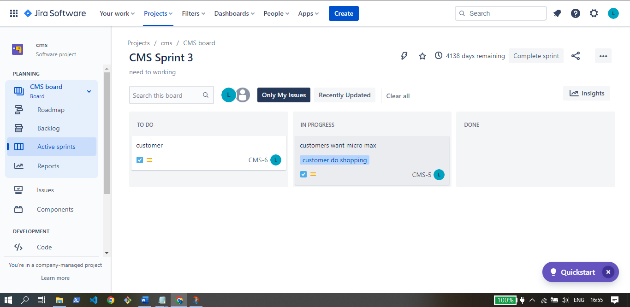
Task:

* project: The name of the project
* issue type dialogue box and now select task option
* summary option is used to type summary about the issue
* Attachment: attach the file from browse or any file required
* Description is used type the large description about the project issue
* and having assign option it helps to assign issue to some one
* and epic link option its used connected last epic
* sprint option to select active sprint
* and click on create



Active Sprints:

* Now go to the project dash board
* Now click on backlog and new add issues like task
* and now click on create sprint
* initialized the timing about to starting and ending of the project and click on create
* now click on Active sprints
* its contains to do/ in progress/ done options
* its shows the issues doing in the process it in the to column or processing of the project it shows in the in prograss column and its completed then drag to task into the done column



7. Write various stages of Agile Methodology?

Stages:

1. Prepare epic & stories
2. Prepare team
3. Sprint plan
4. Scrum call /daily

Prepare epic & Stories:

An epic is a large body of work that can be broken down into a number of smaller stories, or sometimes called “Issues” in Jira. Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards. Epics are almost always delivered over a set of sprints.

Epic in Agile Methodology is a big chunk of work which can be divided into smaller user stories. An Epic can be spread across sprints and even across agile teams. An Epic can be a high-level description of what the client wants, and accordingly, it has some value attached to it

Epic is stories are created by business analyst.

Prepare Scrum Team:

a Scrum Team includes five to eleven people who share the various tasks and responsibilities related to the delivery of the project/product. It is a group of self-motivated individuals who work collaboratively towards successful product delivery.

A high level of communication is expected between the Scrum Team members to ensure they are focused on the same goal while maintaining mutual respect throughout the process.

scrum team follows below rules

Transparency:

Everyone in the team will have an easy and transparent flow of information about the common goal and the roles and responsibilities of each individual.

Inspection:

All team members are entitled to do timely checks on the progress towards a common goal.

Adaptation :

An agile Scrum Team adapts to changes as soon as possible to optimize the product value.

The Scrum Team roles are categorized into three – Scrum Master, Product Owner, and the Development Team. Each of the roles has a specific set of responsibilities throughout the project management cycle, although they are closely interrelated.

Scrum master:

The name was initially used to depict someone who has expertise in the Scrum framework so that they can teach others. In simple terms, the Scrum Master can be defined as the servant-leader of the Scrum Team. The individual is responsible for ensuring that the team adheres to the theory, practices, and rules of Scrum

* Maximizing the value created by the Scrum Team
* Setting the stage for the Scrum Team to work collaboratively and effectively
* Mentoring the team to comply with the agile principles
* Motivating and influencing at tactical and strategic levels

Create Sprint Plan:

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

A good sprint plan motivates everyone by defining an outcome and a clear plan for success. But be careful planning too upfront. Instead of building the most complete, “every minute of the sprint is accounted for” sprint plan, focus on the goal and build enough of a sprint backlog to get started. Next, ensure that the product backlog is ordered to allow the team to pick up work if they delivered on the sprint goal early.

* Sprint planning should be constrained no more than two hours for each week of the sprint.

So, for example, the sprint planning meeting for a two-week sprint would be no longer than four hours. This is called "timeboxing", or setting a maximum amount of time for the team to accomplish a task, in this case, planning the sprint.

Scrum call /Daily:

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint’s work.

The Scrum Master ensures that the meeting happens, but the Developers are responsible for conducting the Daily Scrum. The Scrum Master teaches them to keep the Daily Scrum within the 15-minute to 30 mintues.

8. Difference between water fall model , Agile and Devops?

|  |  |  |
| --- | --- | --- |
| **Waterfall model** | **Agile** | **Devops** |
| In general it methodology gets separated into several stages. | Sprints are used breakdown  The project into manageable stages | In the devops continuous  Deployment of software |
| The waterfall technique is a sequential design process | Agile model follows the concept of consistent growth during the project it self so that after word reduces the risk of complexity the requirements. | The deployment comprises analysing the requirements, designing, developing and testing of the software components or frame work |
| It developed the single application | Focus on the development of single application | A corporate wide approach to software development |
| Gantt charts are used | Scrum, Kanban, scrum bar  Lean | Jenkins is used in the devops |
| Waterfall model is established in 1970 by Winston w. Royce | Agile is developed in spring 2000 by the group of were ken schwaber and jeff sutherland | Devops is developed in 2009 by Patrick Debois |

Difference between scrum and Kanban

|  |  |
| --- | --- |
| **Kanban** | **Scrum** |
| No defined roles in Kanban | Scrum master / product owner and developer team roles are contained in the scrum |
| Continuous delivery cycle | Sprint delivery cycle lasts one of the four weeks |
| change policy can be incorporated by time | Change policy generally not made during sprints |
| Artifacts Kanban board | Product back log , Sprint backlog ,product management |
| Jira software , kanbanize , swift Kanban, trello , asana tools are used in Kanban | Jira software, axo soft, vivify scrum, target process tools are used in scrum |
| Effective, efficient, predictable are pillars in Kanban | Transparency, adaption, inspection |

Difference between git and bit bucket:

|  |  |
| --- | --- |
| **Git** | **Bit bucket** |
| Written c , shell, perl, Tcl and python | Written in Python |
| Developed by Linux Torvalds | Developed by Jesper Noehr |
| A distributed version control system for tracking changes in source code during the software development | A web based version control repository hosting service owned by Atlassian |
| Allows development to track the changes made to the source code during the software development process | Provides web hosting service for git and mercurial repositories. |

Difference between local version control system, centralized version control system & distributed version control system:

|  |  |  |
| --- | --- | --- |
| **Local version control system** | **Centralized version control system** | **Distributed version control system** |
| Local version control system is a local data base located on your local computer | Centralized version control system is simplest form version control in which the central repository of the server provides the latest code to client machines | Distributed version control system is the form of version where the complete code base (including its full history) is mirrored on every developer computer |
| It contains file and everything is stored in locally | There are no local repositories | There are local repositories |
| No need internet connections | Always required internet connectivity | Developer can work with an local repository with an internet connection. |

Difference between dos and windows:

|  |  |
| --- | --- |
| **Dos** | **Window** |
| Dos stands for disk operating system | Windows stands for windows operating system |
| Dos single taking operating system | Windows multiple taking operating system |
| Dos consumes quite low power | Windows consumes high power |
| Dos has no support for networking | Windows supports networking |
| Dos based on command line interface | Windows based on graphical user interface |
| Dos not support multimedia | Windows supports multimedia likes games, videos , audies etc |
| Commands execution is faster than windows | Operations are slower as compared to dos |

9. Write names of 15 devops tools?

1. Katalon Testops
2. Kubernetes
3. Docker
4. Katalon Studio
5. Ranorex Studio
6. Jenkins
7. Azure Devops
8. Ansible
9. Chef
10. Git
11. Terraform
12. Gradle
13. Jira
14. Trello
15. Raygun

10.Write names of 10 cloud providers?

1. Amazon Web Services
2. Microsoft Azure
3. Google Cloud Platform
4. Alibaba Cloud
5. Salesforce
6. Ibm
7. Digital Ocean
8. Dell
9. Adobe
10. Dropbox
11. Cloud Contact AI