

Day 4 : Assignment

Q.What is Git ?

A. --> Git is an open-source distributed version control system.

-->It is designed to handle minor to major projects with high speed and efficiency.

--> It is developed to co-ordinate the work among the developers.

-->The version control allows us to track and work together with our team members at the same workspace.

-->Git is foundation of many services like GitHub and GitLab, but we can use Git without using any other Git services.

-->Git can be used privately and publicly.

-->Git was created by Linus Torvalds in 2005 to develop Linux Kernel.

-->It is also used as an important distributed version-control tool for the DevOps.

--> Git is easy to learn, and has fast performance.

-->It is superior to other SCM tools like Subversion, CVS, Perforce, and ClearCase.

Features of Git:

>>Open Source

>>Scalable

>>Security

>>Speed

Why use Git:

..Synchronous development:

Everyone has their local instance of the code, and everyone can work on their branches at the same time. Git works offline because almost all operations run locally.

..Increase team speed and productivity:

Git makes keeping track of changes to your code easy for your team. So you can focus on writing code instead of wasting time tracking and merging different versions into your team. It also calculates and stores your main repository locally, making it faster for most operations.

..Open Source:

Open source allows developers from all over the world to contribute to the software and make it more and more powerful through features and extra plugins. This has led to the Linux kernel consisting of 15 million lines of code.

..Security:

SHA-1 cryptography keeps you safe. This algorithm securely manages your versions, files and directories to avoid any damage to your work.

..Git is an Industry Standard:

It is highly popular, and major IDEs support it.

Q. What is GitHub?

A. -->GitHub is a Git repository hosting service.

-->GitHub also facilitates with many of its features, such as access control and collaboration.

-->It provides a Web-based graphical interface.

-->GitHub is an American company.

-->It hosts source code of your project in the form of different programming languages and keeps track of the various changes made by programmers.

-->It offers both distributed version control and source code management (SCM) functionality of Git.

-->It also facilitates with some collaboration features such as bug tracking, feature requests, task management for every project.

Features of GitHub:

--GitHub is a place where programmers and designers work together.

--They collaborate, contribute, and fix bugs together.

--It hosts plenty of open source projects and codes of various programming language

--Some of its significant features are as follows

>>Collaboration

>>Integrated issue and bug tracking

>>Graphical representation of branches

>>Git repositories hosting

>>Project management

>>Team management

>>Code hosting

>>Track and assign tasks

>>Conversations

>>Wikisc

Q. Difference between git and gitHub ?

A. Git :

1.It is a software .

2.It is installed locally on the system.

3.It is a command line tool.

4.It is a tool to manage different versions of edits,made to files in a git repostiory.

5.It provides functionalities likeVersion Control,system source management.

6.It is a Maintained by Linux.

7.It is a Open-Source.

8.It is a No user management.

9.Little to no competition.

GitHub:

1.It is a service.

2.It is hosted on Web.

3.It provides a graphical interface.

4.It is a space to upload a copy of the git repository.

5.It provides functionalites of Git like VCS,source code managemen as well as adding few of its own features.

6.It is a Maintained by Microsoft.

7.It is a Free or paid membership.

8.Built-in user management.

9.High competition.

Q. What is Deployment ?

A. Deployment is the mechanism through which applications, modules, updates, and patches are delivered from developers to users.

The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changes in customer preferences or requirements and the quality of each change.

Deployment Process:

>>Requirements Analysis

>>Application Designing

>>Application Development

>>Application Testing
>>Maintenance & Support

Q. What is difference between Google Drive & GitHub ?

>>Google Drive is a free service from Google that allows you to store files online and access them anywhere using the cloud.

Google Drive also gives you access to free web-based applications for creating documents, spreadsheets, presentations, and more.

>>GitHub is a code hosting platform for collaboration and version control. GitHub lets you (and others) work together on projects.

Q. What is JIRA?

A. >>JIRA is a project management software developed by the Australian company Atlassian.

>>The word JIRA is derived from the Japanese word 'Gozilla', meaning Godzilla.

>>The software is based on agile methodology.

>>If you're wondering what is jira used for, the answer is multiple purposes - bug tracking, issue tracking, and project management.

>>Many businesses also use JIRA software in non-standard ways as warehouse automation tool, document flow, expenses optimization, and others.

>>The JIRA dashboard contains several useful functions and features which enable easy handling of issues.

>>One of the most sought after agile project management solutions, JIRA has recently tweaked some of its products for all kinds of teams and organizations including IT, marketing, operations, finance, HR, legal and other departments.

Q. What is Agile Model

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development.

Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process.

Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks.

The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Phases of Agile Model:

Following are the phases in the Agile model are as follows:

>>Requirements gathering

>> Design the requirements

>>Construction/ iteration

>>Testing/ Quality assurance

>> Deployment

>> Feedback

1. Requirements gathering:

In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

2. Design the requirements:

When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3. Construction/ iteration:

When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4. Testing:

In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

5. Deployment:

In this phase, the team issues a product for the user's work environment.

6. Feedback:

After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

Q. What is Waterfall Model ?

A. Winston Royce introduced the Waterfall Model in 1970.

This model has five phases:

- >> Requirements analysis
- >> specification,
- >> design,
- >> implementation,
- >> unit testing,
- >> integration ,
- >> system testing,
- >> operation ,
- >> maintenance.

The steps always follow in this order and do not overlap.

The developer must complete every phase before the next phase begins.

This model is named "Waterfall Model", because its diagrammatic representation resembles a cascade of waterfalls.

Q. What is SDLC ?

A. The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time.

The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.

Stages of the Software Development Life Cycle:

SDLC specifies the task(s) to be performed at various stages by a software engineer or developer.

It ensures that the end product is able to meet the customer's expectations and fits within the overall budget.

Hence, it's vital for a software developer to have prior knowledge of this software development process.

The SDLC model involves six phases or stages while developing any software.

SDLC is a collection of these six stages, and the stages of SDLC are as follows:

Stage-1: Planning and Requirement Analysis

Planning is a crucial step in everything, just as in software development.

In this same stage, requirement analysis is also performed by the developers of the organization.

This is attained from customer inputs, and sales department/market surveys

Stage-2: Defining Requirements

In this stage, all the requirements for the target software are specified.

These requirements get approval from customers, market analysts, and stakeholders.

This is fulfilled by utilizing SRS (Software Requirement Specification).

This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

Stage-3: Designing Architecture

SRS is a reference for software designers to come up with the best architecture for the software.

Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

Stage 3: Design

Stage-4: Developing Product

At this stage, the fundamental development of the product starts.

For this, developers use a specific programming code as per the design in the DDS.

Hence, it is important for the coders to follow the protocols set by the association.

Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage.

Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.

Stage 4: Development

Stage-5: Product Testing and Integration

After the development of the product, testing of the software is necessary to ensure its smooth execution.

Although, minimal testing is conducted at every stage of SDLC.

Therefore, at this stage, all the probable flaws are tracked, fixed, and retested.

This ensures that the product confronts the quality requirements of SRS.

Stage-6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization's strategy.

Then it is tested in a real industrial environment.

It is important to ensure its smooth performance.

If it performs well, the organization sends out the product as a whole.

After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers.

However, this alone is not enough. Therefore, along with the deployment, the product's supervision.