

1.what is git?

Git is a version control system that helps you track changes in your files and collaborate with others on projects. It allows multiple people to work on the same files simultaneously, keeps a history of all changes made to the files, and provides mechanisms to manage conflicts and revert to previous versions if needed. Essentially, Git helps you keep your project organized, coordinated, and efficient.

2.what is git hub and deployment?

GitHub is an online platform where developers store, share, and collaborate on code. It facilitates teamwork, tracks changes, and fosters open-source contributions, making coding projects more accessible and efficient for everyone involved.

The deployment phase is the final step in the software development life cycle and delivers the final product to the customer in a live production environment. After the product deploys, the product is ready for customers to use.

3.what is the difference between git hub and google drive?

GitHub is a place to share code with friends, co-workers, classmates, and complete strangers, helping individuals and teams to write faster, better code

Google Drive is a cloud storage and backup platform to access files, docs, photos & more, store them in a safe place, and collaborate with other people

4.what is jira?

In simple terms, Jira is a software tool used for project management and issue tracking. It helps teams organize their work, track tasks, and collaborate on projects. Jira allows users to create and prioritize tasks, assign them to team members, set deadlines, and track progress through various stages of completion. It's widely used in software development but can also be utilized in other industries for managing projects and workflows.

5.What is sdlc?

In simple terms, SDLC stands for Software Development Life Cycle. It's a structured process that software developers use to design, develop, test, and deploy software applications.

Here's a breakdown of the main stages of the SDLC:

1. **Planning:** This is where the project scope, requirements, and objectives are defined. Developers and stakeholders decide what the software needs to do and how it will be implemented.
2. **Analysis:** During this stage, the existing system (if any) is studied and potential improvements are identified. Requirements are gathered from stakeholders to understand what the software should accomplish.

3. **Design:** In this phase, developers create a detailed plan for how the software will be built. This includes architectural design, database design, user interface design, and more.
4. **Implementation:** This is where the actual coding of the software takes place based on the design specifications. Developers write the code according to the requirements and design documents.
5. **Testing:** Once the software is coded, it undergoes testing to find and fix any defects or bugs. Testing ensures that the software meets the requirements and functions correctly.
6. **Deployment:** After testing, the software is deployed or released to users. This may involve installing it on servers, distributing it to users, or making it available for download.
7. **Maintenance:** Once the software is in use, it requires ongoing maintenance and support. This includes fixing bugs, making updates, and addressing any issues that arise during operation.

The SDLC provides a systematic approach to software development, helping teams manage the process from start to finish and ensuring that the final product meets the needs of users.

5. what is waterfall methodology?

In simple terms, the Waterfall methodology is a traditional approach to software development where the process flows steadily downwards, like a waterfall, through several distinct phases. Each phase must be completed before moving on to the next one, and changes are not easily accommodated once a phase is finished.

Here's a breakdown of the main phases of the Waterfall methodology:

1. **Requirements:** This is where the project requirements are gathered and documented. It involves understanding what the software needs to do and what features it should have.
2. **Design:** In this phase, the overall architecture and design of the software are created based on the requirements gathered in the previous phase. This includes designing the user interface, database structure, and other technical aspects.
3. **Implementation:** Once the design is complete, developers start writing the code for the software based on the design specifications.
4. **Testing:** After the code is written, the software is tested to ensure that it works correctly and meets the requirements. Testing typically involves both automated and manual testing to identify any defects or bugs.
5. **Deployment:** Once the software has been thoroughly tested and approved, it is deployed or released to users. This may involve installing it on servers, distributing it to users, or making it available for download.
6. **Maintenance:** After deployment, the software requires ongoing maintenance and support. This includes fixing bugs, making updates, and addressing any issues that arise during operation.

The Waterfall methodology is characterized by its sequential approach, where each phase relies on the completion of the previous one. While it can provide a clear and structured process, it can also be rigid and less adaptable to changes compared to more modern agile methodologies.

6. What is agile model?

The agile model refers to the iterative approach to delivering a software product. This means that instead of delivering a large project only when all parts are complete, a team breaks down a large project into smaller parts, and delivers these completed smaller parts in regular cycles.