

What Is the Software Development Life Cycle, and How Does It Relate to Testing?

(🔗) freecodecamp.org/learn/front-end-development-libraries-v9/lecture-understanding-the-different-types-of-testing/what-is-the-software-development-life-cycle-and-how-does-it-relate-to-testing



When working on software teams, it is important to have a process for designing and building out new products and services. The Software Development Life Cycle is commonly used by software teams to build and deploy applications.

In this lesson, we will break down the different stages of the Software Development Life Cycle (SDLC) and the role that testing plays in it.

Here are the standard stages used in the Software Development Life Cycle:

- Planning stage
- Design stage
- Implementation stage
- Testing stage
- Deployment stage and
- Maintenance stage

Let's take a closer look at all of those stages in more detail.

In the planning stage, the development team collects requirements for the proposed work from the stakeholders. A stakeholder is someone with a vested interest in the success of the product or service. Examples of stakeholders could be investors, owners, or even other employees. The planning phase also includes estimating costs for the work and creating a schedule for when the work will be completed.

The next stage is the Design stage. In this phase, the software team breaks down the requirements and decide on the best approaches for solutions. This includes deciding on which external tools can be used to help things along and what internal tooling needs to be built to get the job done.

After the design stage, you have the implementation stage. This involves the software team breaking down the requirements into manageable tasks that they divide up amongst themselves and build out.

Once the implementation stage is complete, the software team moves into the testing stage. This involves manual and automated testing for the new work. Sometimes the team tests out the application throughout the entire development stage to catch and fix any issues that come up.

The next stage is the deployment stage. This is where the team deploys the new changes to a build or testing environment. This allows the team to test out the application some more and fix any issues before the changes go to production. The production application is the app that end users will use.

The last stage is the maintenance stage. This involves fixing any issues that arise from customers in the production application. This also involves monitoring the system for any possible improvements on the software.

Finally, there are different models of the Software Development Life Cycle used in the industry, such as Waterfall and Agile. The Waterfall model is where each phase of the life cycle needs to be completed before the next phase can begin. In contrast, the Agile model focuses on iterative development by breaking down work into sprints so teams can quickly make changes and deliver work over time.

By going through the Software Development Life Cycle, the development team and stakeholders can be on the same page when it comes to requirements, cost estimates, and schedules. This makes for more efficient collaboration and communication across the board.

Questions

Which of the following is NOT a stage used in the Software Development Life Cycle?

Planning Stage

Testing Stage

Regression Stage

Deployment Stage

What is the design stage?

This is where the team will collect requirements and deadlines from stakeholders.

This is where the team will break down the requirements and design solutions.

This is where the team will only focus on UI development.

This is where the team will only create Figma and Sketch mockups of the work to be done.

What does the Agile model focus on?

Iterative development

Slow development

Expensive development

UI/UX development

Navigated to What Is the Software Development Life Cycle, and How Does It Relate to Testing?