Documentation DevOps

To compare Software Engineering Methodology with DevOps

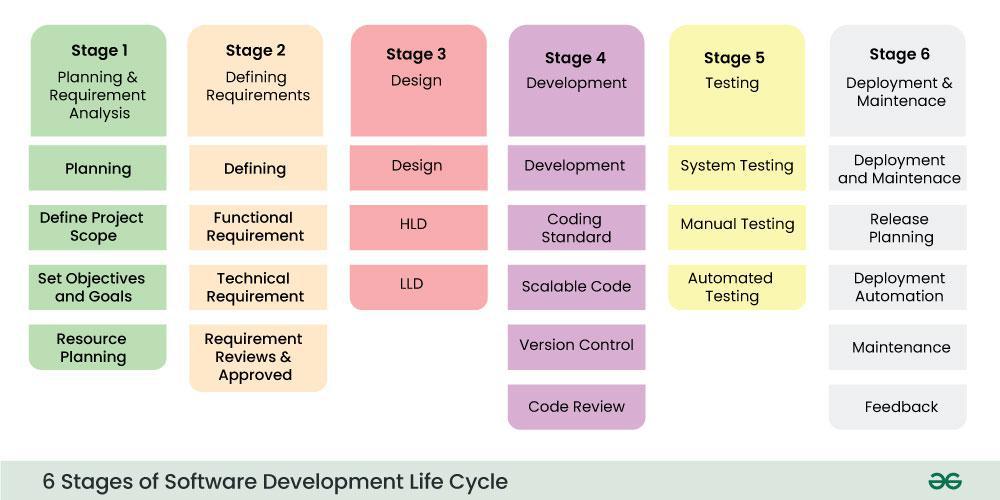
SDLC:

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step. The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user's requirements. SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets user’s requirements. In this article we will see Software Development Life Cycle (SDLC) in detail.



Stages of the Software Development Life Cycle

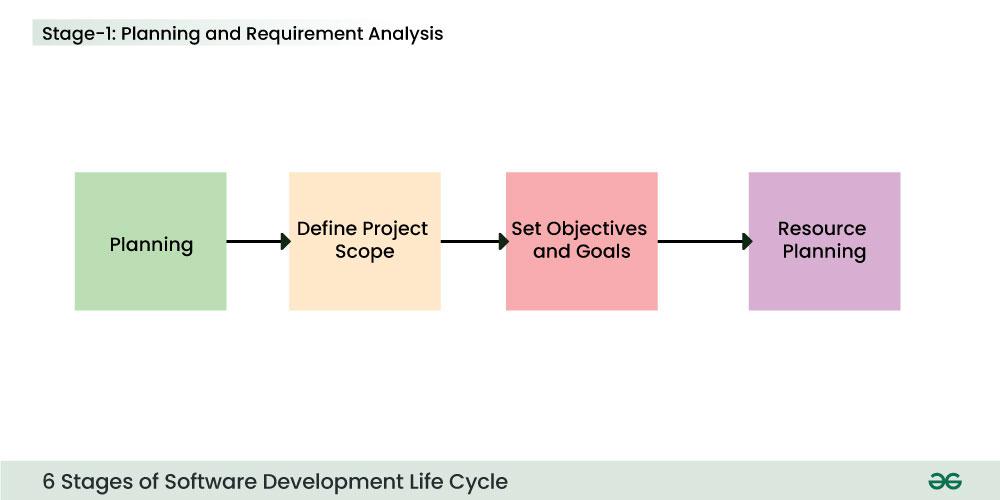
SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer's expectations and fits within the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process. 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.

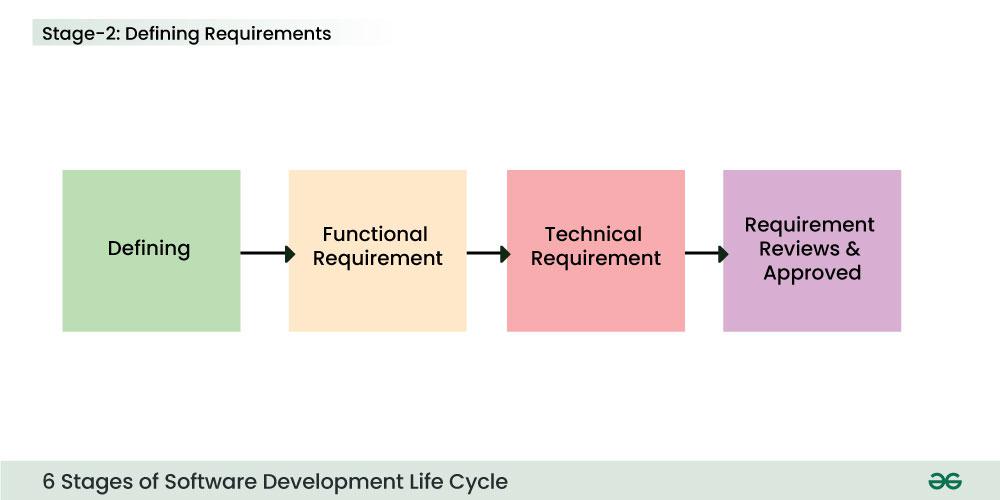
The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.



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.

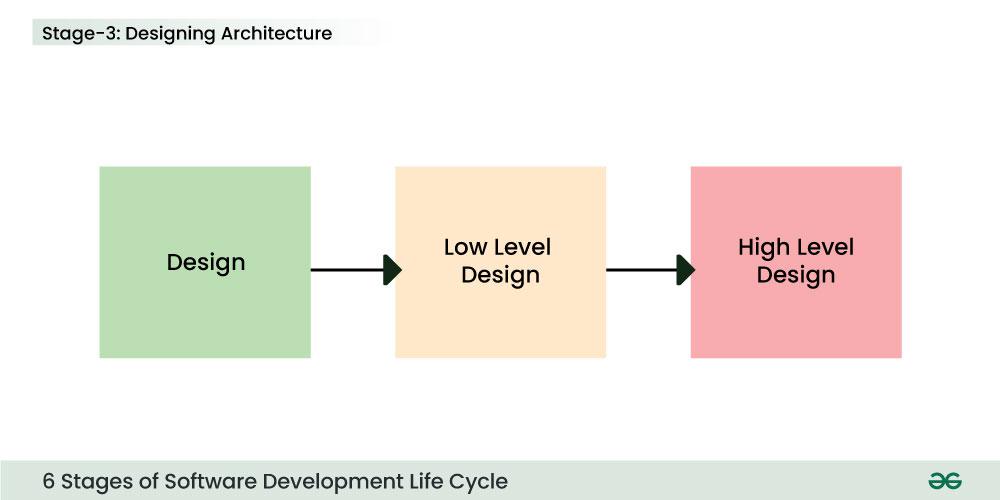
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.



Stage-3: Designing Architecture

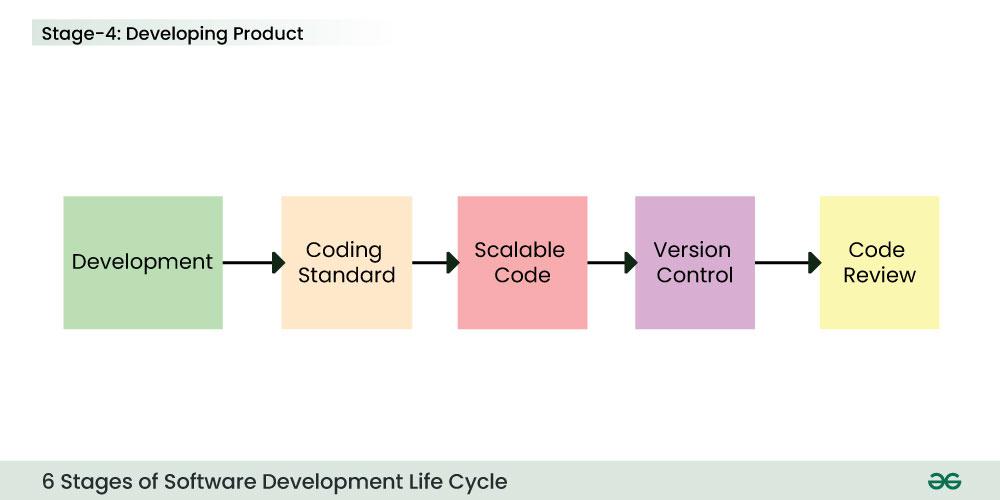
SRS is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.



Stage-4: Developing Product

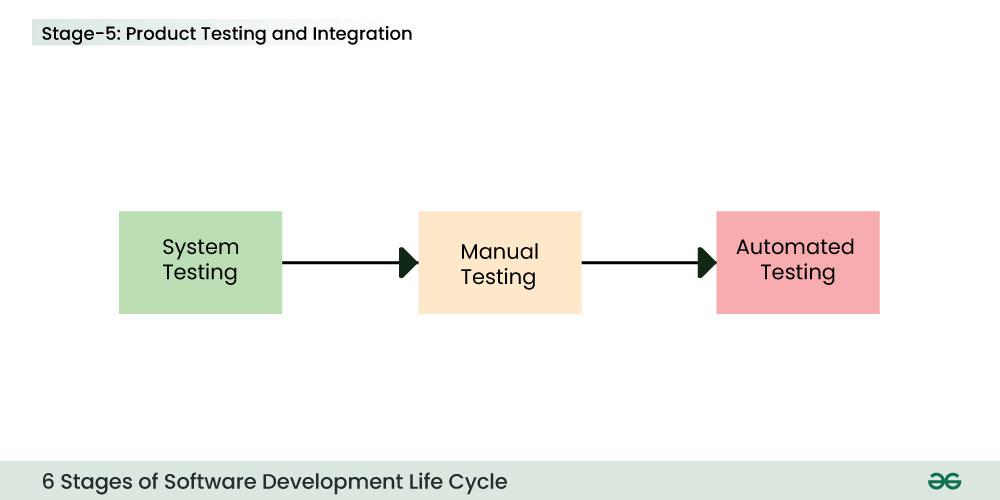
At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, 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-5: Product Testing and Integration

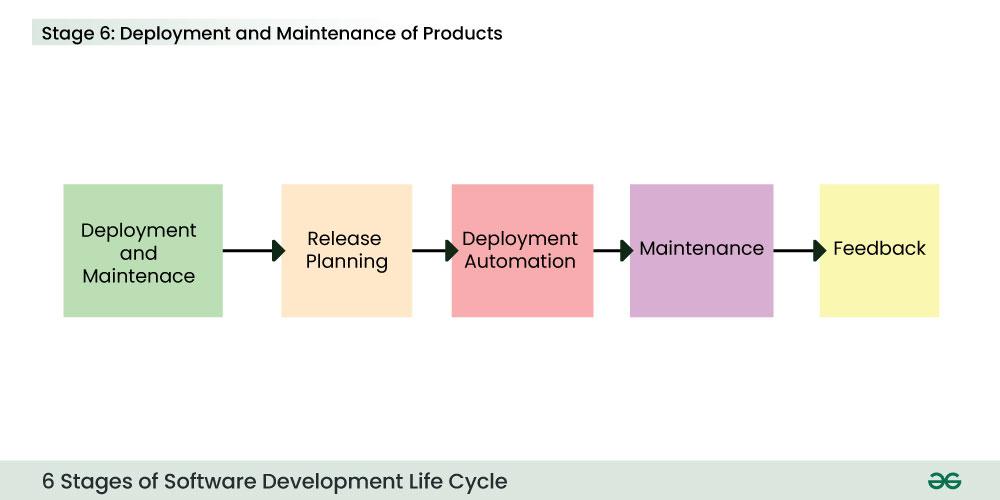
After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

Documentation, Training, and Support: Software documentation is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions, and maintenance. Documentation also provides information about how to use the product. Training in an attempt to improve the current or future employee performance by increasing an employee's ability to work through learning, usually by changing his attitude and developing his skills and understanding.



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.



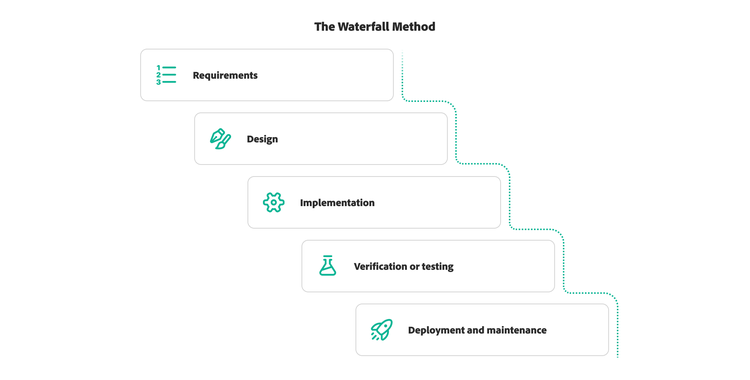
<https://www.geeksforgeeks.org/software-engineering/software-development-life-cycle-sdlc/>

Waterfall Methodology:

The Waterfall methodology is a project management approach that emphasizes a linear progression from the beginning to the end of a project. This methodology, often used by engineers, is front-loaded to rely on careful planning, detailed documentation, and consecutive execution.

The Waterfall methodology — also known as the Waterfall model — is a sequential development process that flows like a waterfall through all phases of a project (for example, analysis, design, development, and testing), with each phase completely wrapping up before the next phase begins.

It is said that the Waterfall methodology follows the adage to “measure twice, cut once.” The success of the Waterfall method depends on the amount and quality of the work done on the front end, documenting everything in advance, including the user interface, user stories, and all the features’ variations and outcomes.



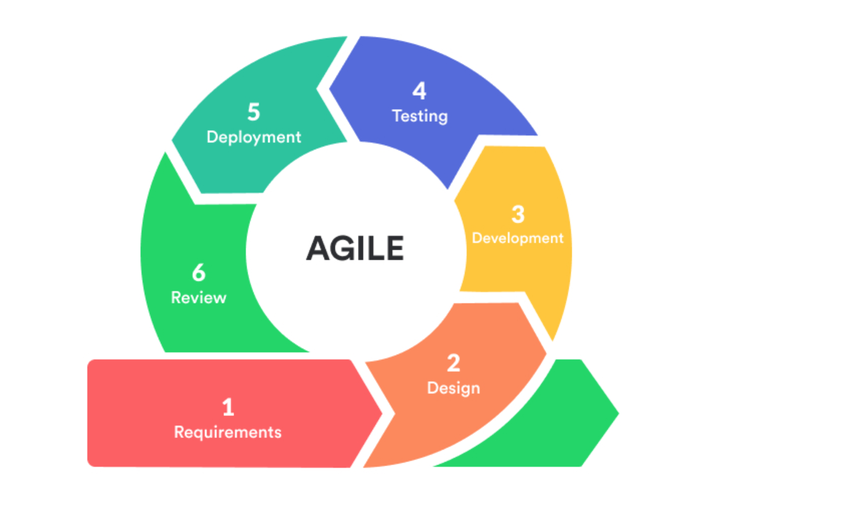
<https://business.adobe.com/blog/basics/waterfall>

Agile Methodology:

Agile methodology is a pivotal project management approach predominantly applied in software development. It embodies a dynamic process where demands and solutions evolve through the synergistic collaboration of self-organizing, cross-functional teams, and their engaged customers.

The Agile methodology is rooted in a set of valuing adaptability and flexibility. Agile stands as a transformative force, enhancing responsiveness to ever-changing business needs.

It empowers teams to deliver tangible progress in incremental, workable increments, fostering a climate of continuous improvement.



**Agile vs Waterfall Methodology: A Quick Comparison**

| **Criteria** | **Agile** | **Waterfall** |
| --- | --- | --- |
| Adaptability | Extremely adaptable, allowing quick responses to changes and evolving technology. | More rigid structure, best suited for projects with a clear and unchanging vision. |
| Project Timeline | Flexible timeline dependent on project development. | Fixed timeline planned from the start. |
| Project Phases | Concurrent work on phases with tight deadlines, team-driven direction. | Linear progression through defined stages, driven by project manager. |
| Flexibility in Direction | Allows for changes even late in the process, suitable for evolving projects. | Less flexibility due to a predefined and unchanging vision. |
| Budget Flexibility | Budget subject to change as project direction evolves. | Less flexible budget planned from the start. |
| Ideal for | Software development where technology evolves rapidly. | Projects with a clear and specific vision that won't change. |
| Stakeholder Feedback | Continuous stakeholder feedback is incorporated throughout. | Deliverables for each stage are clearly defined before moving on. |

DevOps:

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model, where developers who wrote code worked apart from operations who deployed and supported the code. The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.

How does DevOps work?

A DevOps team includes developers and IT operations working collaboratively throughout the product lifecycle, in order to increase the speed and quality of software deployment. It’s a new way of working, a cultural shift, that has significant implications for teams and the organizations they work for.

Under a DevOps model, development and operations teams are no longer “siloed.” Sometimes, these two teams merge into a single team where the engineers work across the entire application lifecycle — from development and test to deployment and operations — and have a range of multidisciplinary skills.

DevOps teams use tools to automate and accelerate processes, which helps to increase reliability. A DevOps toolchain helps teams tackle important DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.

DevOps values are sometimes applied to teams other than development. When security teams adopt a DevOps approach, security is an active and integrated part of the development process. This is called DevSecOps.

<https://zenkit.com/en/blog/agile-methodology-an-overview/>