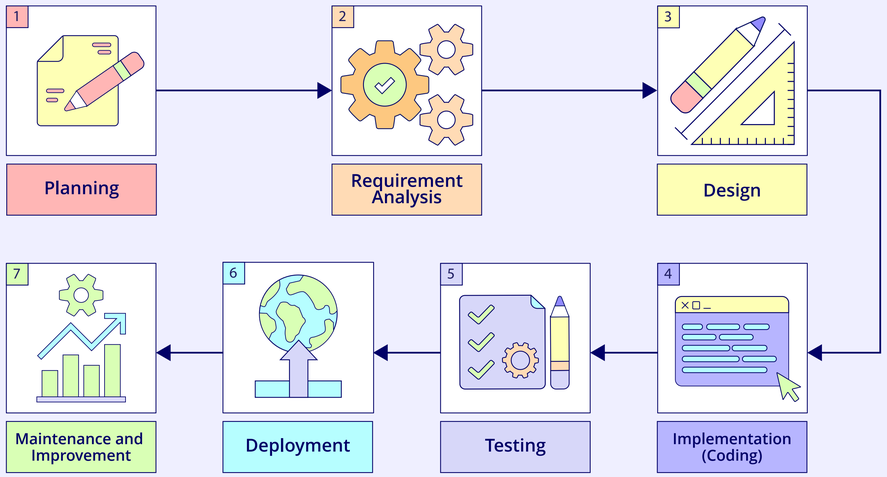
Software Development Life Cycle (SDLC)- CS101

educative

THE 7 PHASES OF SDLC



1 Planning

This phase involves defining the project scope, setting goals, estimating cost and effort, and identifying risks and benefits.

2 Requirement Analysis

This phase focuses on functional requirements and performs solution analysis. It involves communication between stakeholders, including clients, end-users, and developers, to collect and document project requirements.

3 Design

This phase involves creating a blueprint for the software, specifying how different components will interact. After the completion of the design specification, all stakeholders will examine the plan and offer their feedback and suggestions.

The design phase in the SDLC typically encompasses several of the design types, but generally, the following types of design activities are conducted:

- **System design:** This design phase often includes high-level system design, where the overall architecture and components of the system are defined. This phase lays the foundation for subsequent detailed design activities.
- **Architectural design:** This involves making decisions about the overall structure of the system, including the identification of key components, their interactions, and the system's high-level design.
- **Database design:** Database design is an integral part of the design phase, where decisions about the structure of the database, tables, relationships, and data integrity are made.
- **Software design:** Detailed software design is a crucial part of the design phase. It involves creating specifications for the software components, including data structures, algorithms, and interfaces.
- **User Interface (UI) design:** This design phase includes planning and creating the user interface, ensuring that it meets user requirments and provides a positive user experience.

4 Implementation (Coding)

This phase involves the actual coding or programming of the software based on the design specifications. The development team writes code by following the coding standards and guidelines.

5 Testing

This phase is crucial for identifying and fixing bugs or defects in the software. It includes unit testing, integration testing, system testing, and user acceptance testing to ensure the software functions correctly and meets requirements.

6 Deployment

Once the software has been thoroughly tested and approved, it is released or deployed to the production environment for end users.

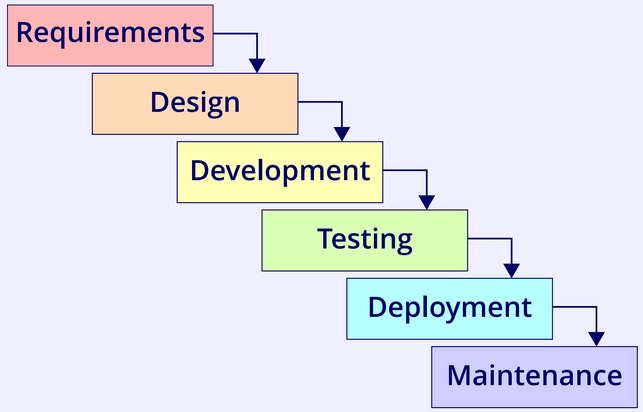
7 Maintenance and Improvement

This phase involves ongoing maintenance and support for the software. It includes fixing bugs, addressing issues, and implementing updates or enhancements as needed throughout the software's lifecycle.

Software Development Life Cycle (SDLC)- CS101

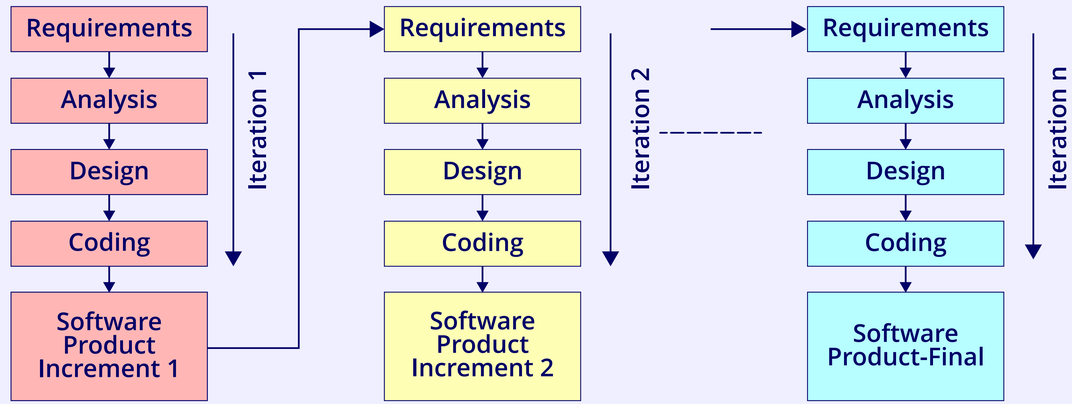
educative

WATERFALL MODEL



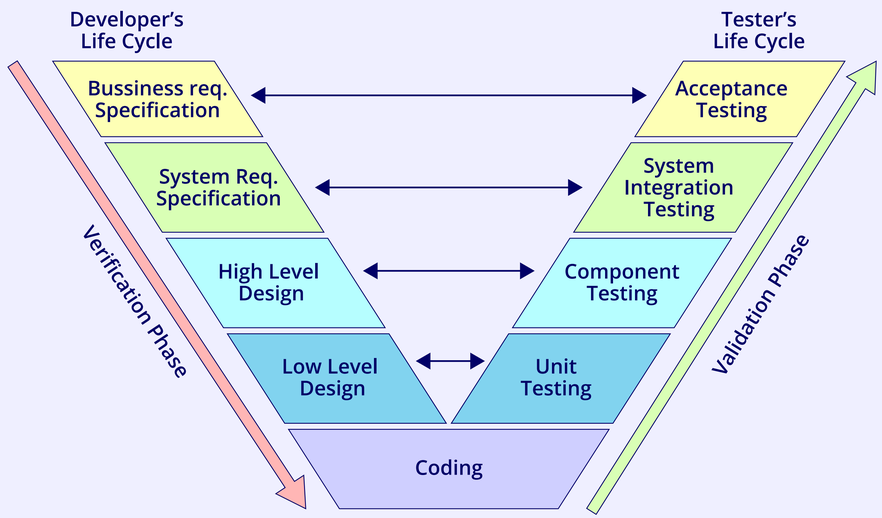
This is a linear and sequential model where progress is seen as flowing steadily downward through several phases, such as requirements, design, implementation, testing, deployment, and maintenance. Once a phase is completed, the development process moves to the next phase.

ITERATIVE / INCREMENTAL MODEL



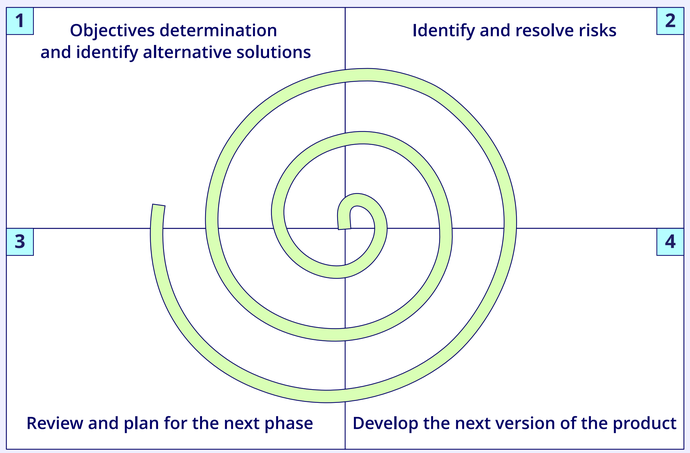
In this model, the software development process is divided into small iterations or increments, with each iteration building upon the previous one. Each iteration typically includes planning, design, implementation, and testing phases.

V - MODEL



The V-Model is an extension of the waterfall model, where each development stage corresponds to a testing phase. The left side of the "V" represents the development phases, and the right side represents the testing phases.

SPIRAL MODEL

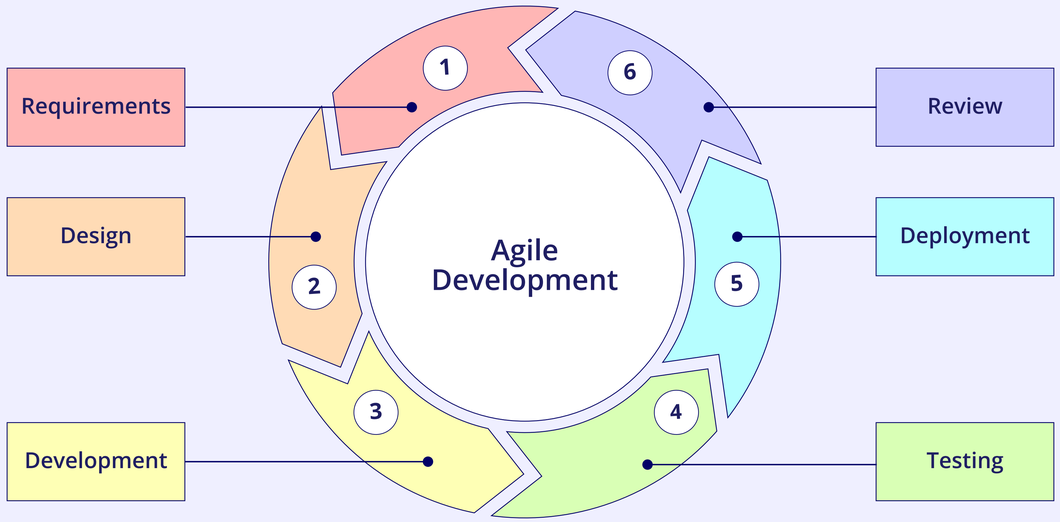


The spiral model combines elements of both the waterfall and iterative models. It includes repeated cycles of planning, risk analysis, engineering, testing, and evaluation of each iteration. This model is particularly useful for large and complex projects.

Software Development Life Cycle (SDLC)- CS101

educative

AGILE MODEL



Agile is an iterative and incremental model that emphasizes flexibility and customer satisfaction. It involves collaboration among cross-functional teams and encourages adaptive responses to changes throughout the development process. Scrum, Kanban, and Extreme Programming (XP) are popular frameworks within the Agile methodology.

COMPARISON OF SDLC MODELS

Aspect	Waterfall Model	Iterative/Incremental Model	V-Model	Spiral Model	Agile Model
Development approach	Sequential and linear	Iterative and incremental	Sequential with testing emphasis	Iterative with risk analysis	Iterative and incremental
Flexibility	Low	High	Low	High	High
Result delivery	Slow	Fast	Slow	Slow	Fast
Risks	High	Low	High	Low	Low
Team size	Large	Large	Any	Large	Small