

## Drawbacks of Waterfall Model

The Waterfall Model is one of the earliest software development life cycles (SDLC) models, characterized by its linear and sequential approach. While it provides a clear structure, this rigidity introduces several serious drawbacks that make it unsuitable for many modern software projects.

One of the most critical limitations of the Waterfall Model is its lack of flexibility. Once a phase is completed, revisiting or modifying it becomes extremely difficult. In real-world projects, requirements often evolve due to market changes, user feedback, or technical discoveries. The Waterfall Model assumes that all requirements are fully known at the beginning, which is rarely the case, leading to costly rework or project failure.

Another major drawback is late testing and defect discovery. Testing is performed only after the implementation phase is completed. As a result, bugs, design flaws, or requirement mismatches are detected very late in the development cycle. Fixing such issues at a later stage is not only expensive but can also compromise the overall system architecture.

The Waterfall Model also suffers minimal customer involvement after the requirements phase. Clients or stakeholders typically do not see a working version of the software until the final stages. This increases the risk of building a system that technically meets documented requirements but fails to satisfy actual user needs.

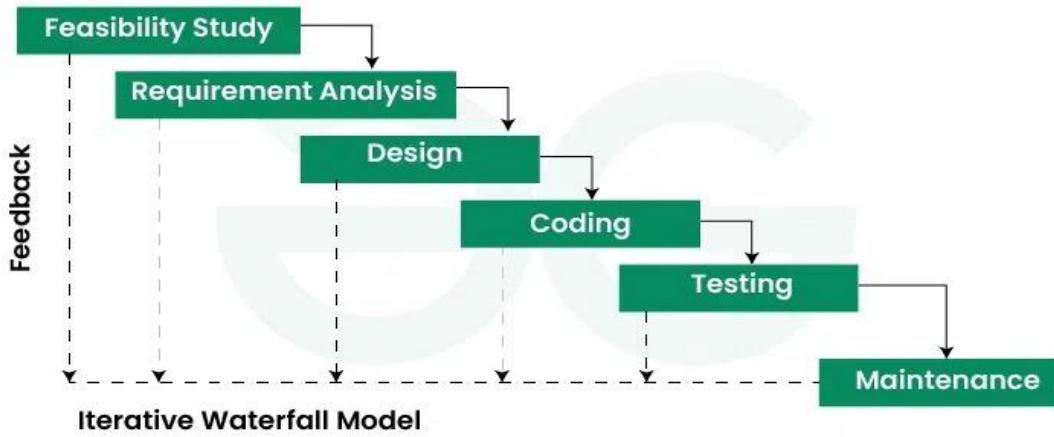
From a project management perspective, the model is high-risk for long and complex projects. Any incorrect assumption made during the early stages propagates throughout the entire lifecycle. Since progress is measured by documentation rather than working software, it can create a false sense of advancement.

Additionally, the Waterfall Model is poorly suited for innovation and experimentation. Modern software development often involves learning through iteration, prototyping, and continuous improvement. The strict phase boundaries of Waterfall discourage experimentation and adaptability, limiting creative problem-solving.

Key drawbacks can be summarized as follows:

- Rigid structure that does not accommodate changing requirements.
- Late detection of errors, increasing cost and effort of fixes.
- Limited user feedback, leading to misaligned final products.

- High dependency on early-stage accuracy, making mistakes is extremely costly.
- Not suitable for dynamic or evolving projects.



Phases of Iterative Waterfall Model

