SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

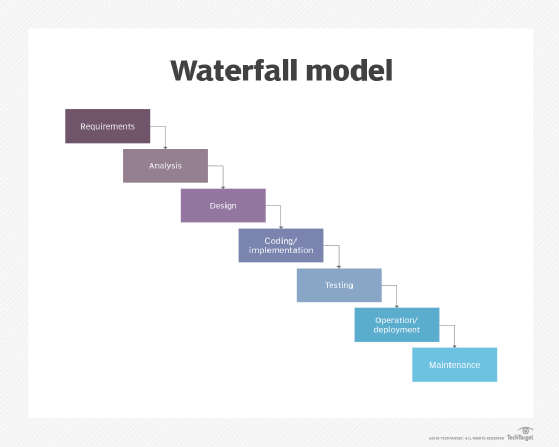
The waterfall model is a linear, sequential approach to the [software development lifecycle](https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle) (SDLC) that is popular in software engineering and product development.

The waterfall model uses a logical progression of SDLC steps for a project, similar to the direction water flows over the edge of a cliff. It sets distinct endpoints or goals for each phase of development. Those endpoints or goals can't be revisited after their completion.

Who uses the waterfall model?

Project teams and Project managers use the waterfall model to achieve goals based on the needs of their business. The model is used in many different project management contexts, such as in construction, manufacturing, IT and software development.

WATERFALL MODEL OF (SDLC)



RequirementGathering

System Analysis

Design

Coding/ Implementation

Testing

Deployment

Maintenance

**Requirements.** Potential requirements, deadlines and guidelines for the project are analyzed and placed into a formal requirements document, also called a [*functional specification*](https://www.techtarget.com/searchsoftwarequality/definition/functional-specification). This stage of development defines and plans the project without mentioning specific processes.

**Analysis.** The system specifications are analyzed to generate product models and [business logic](https://www.techtarget.com/whatis/definition/business-logic) to guide production. This is also when financial and technical resources are audited for feasibility.

**Design.** A design specification document is created to outline technical design requirements, such as the programming language, [hardware](https://www.techtarget.com/searchnetworking/definition/hardware), data sources, architecture and services.

**Coding and implementation.** The [source code](https://www.techtarget.com/searchapparchitecture/definition/source-code) is developed using the models, logic and requirement specifications designated in the prior phases. Typically, the system is coded in smaller components, or units, before being put together.

**Testing.** This is when [quality assurance](https://www.techtarget.com/searchsoftwarequality/definition/quality-assurance), [unit](https://www.techtarget.com/searchsoftwarequality/definition/unit-testing), [system](https://www.techtarget.com/searchsoftwarequality/definition/system-testing) and [beta](https://www.techtarget.com/whatis/definition/beta-test) tests identify issues that must be resolved. This may cause a forced repeat of the coding stage for [debugging](https://www.techtarget.com/searchsoftwarequality/definition/debugging). If the system passes integration and testing, the waterfall continues forward.

**Operation and deployment.** The product or application is deemed fully functional and is deployed to a live environment.

**Maintenance.** Corrective, adaptive and perfective maintenance is carried out indefinitely to improve, update and enhance the product and its functionality. This could include releasing [patch](https://www.techtarget.com/searchenterprisedesktop/definition/patch) updates and new versions.

**Advantages of the waterfall model**

Today, Agile methodology is often [used in place](https://www.pmi.org/learning/library/agile-versus-waterfall-approach-erp-project-6300) of the waterfall model. However, there are advantages to the waterfall approach, such as the following:

enables large or changing teams to move toward a common goal that's been defined in the requirements stage;

forces structured, disciplined organization;

simplifies understanding, following and arranging tasks;

facilitates departmentalization and managerial control based on the schedule or deadlines;

reinforces [good coding habits](https://www.techtarget.com/searchsoftwarequality/feature/Learn-5-defensive-programming-techniques-from-experts) to define before implementing design and then code;

enables early system design and specification changes to be easily done; and

clearly defines milestones and deadlines.

**Disadvantages of the waterfall model**

Disadvantages of the waterfall model typically center around the risk associated with a lack of revision and flexibility. Specific issues include the following:

Design isn't adaptive; when a flaw is found, the entire process often needs to start over.

Method doesn't incorporate midprocess user or client feedback, and makes changes based on results.

Waterfall model delays testing until the end of the development lifecycle.

It doesn't consider error correction.

The methodology doesn't handle requests for changes, [scope](https://www.techtarget.com/searchcio/definition/project-scope) adjustments and updates well.

Waterfall doesn't let processes overlap for simultaneous work on different phases, reducing overall efficiency.

No working product is available until the later stages of the project lifecycle.

Waterfall isn't ideal for complex, high-risk ongoing projects.