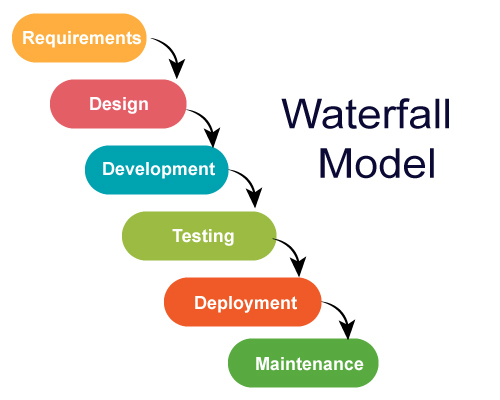
First introduced by Dr. Winston W. Royce in a paper [published in 1970](http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf), the waterfall model is a software development process. The waterfall model emphasizes that a logical progression of steps be taken throughout the software development life cycle (SDLC), much like the cascading steps down an incremental waterfall. While the popularity of the waterfall model has waned over recent years in favor of more agile methodologies, the logical nature of the sequential process used in the waterfall method cannot be denied, and it remains a common design process in the industry.

Throughout this article we’ll examine what specific stages make up the core of the waterfall model, when and where it is best implemented, and scenarios where it might be avoided in favor of other design philosophies.



What is Waterfall Methodology?

Waterfall methodology is a linear project management approach, where stakeholder and customer requirements are gathered at the beginning of the project, and then a sequential project plan is created to accommodate those requirements. The waterfall approach was first conceived by Winston W. Royce in 1970, and it was quickly adopted in a variety of industries due to its logical sequencing and ease of implementation.

Some of the industries that regularly use the waterfall method include construction, IT and software development. However, the term “waterfall” is usually used in a software context, so the rest of this post will concern the waterfall model as it pertains to software development.

The Phases of the Waterfall Model

The waterfall model is a project management methodology that has at least five to seven phases that follow in strict linear order, where a phase can’t begin until the previous phase has been completed. The specific names of the phases vary, but they were originally defined by Royce in the following way:

* Requirements: The key aspect of waterfall is that all customer requirements are gathered at the beginning of the project, allowing every other phase to be planned without further customer involvement until the product is complete. It is assumed that all requirements can be gathered at this phase.
* Design: The design phase is best broken up into logical design and physical design subphases. The logical design subphase is when possible solutions are brainstormed and theorized. The physical design subphase is when those theoretical ideas and schemas are made into concrete specifications.
* Implementation: The implementation phase is when programmers assimilate the requirements and specifications from the previous phases and produce actual code.
* Verification: This phase is when the customer reviews the product to make sure that it meets the requirements laid out at the beginning of the project. This is done by releasing a completed product to the customer.
* Maintenance: The customer is regularly using the product during the maintenance phase, discovering bugs, inadequate features and other errors that occurred during production. The production team applies these fixes as necessary until the customer is satisfied.