



Understanding the pros and cons of the Waterfall Model of software development

by Contributor Melonfire in Developer  on September 22, 2006, 7:08 AM PST

Waterfall development is a software development model involving a phased progression of activities, marked by feedback loops, leading to the release of a software product. This article provides a quick and dirty introduction to the model, explaining what it is, how it's supposed to work, describing the six phases, and why the model can fail.

Say the words "waterfall development" to most people and chances are they're going to be thinking of a bunch of condos under Niagara Falls. Imagine their surprise, then, when you tell them that waterfall development is actually a [software development](https://www.techrepublic.com/5264-1-0.html?query=application%20development)

[model](https://www.techrepublic.com/5264-1-0.html?query=application%20development) which involves a phased progression of activities leading to the release of a software product. This article provides a quick and dirty introduction to the model, explaining what it is, how it's supposed to work, and why it can fail.

Overview

Waterfall development isn't new -- it's been around since 1970 -- but most developers still only have a vague idea of what it means. Essentially, it's a framework for software development in which development proceeds sequentially through a series of phases, starting with system requirements analysis and leading up to product release and maintenance. Feedback loops exist between each phase, so that as new information is uncovered or problems are discovered, it is possible to "go back" a phase and make appropriate modification. Progress "flows" from one stage to the next, much like the waterfall that gives the model its name.

A number of variants of this model exist, with each one quoting slightly different labels for the various stages. In general, however, the model may be considered as having six distinct phases, described below:

1. **Requirements analysis:** This first step is also the most important, because it involves gathering information about what the customer needs and defining, in the clearest possible terms, the problem that the product is expected to solve. Analysis includes understanding the customer's business context and constraints, the functions the product must perform, the performance levels it must adhere to, and the external systems it must be compatible with. Techniques used to obtain this understanding include customer interviews, use cases, and "shopping lists" of software features. The results of the analysis are typically captured in a formal requirements specification, which serves as input to the next step.
2. **Design:** This step consists of "defining the hardware and software architecture, components, modules, interfaces, and data...to satisfy specified requirements" ([Wikipedia \(http://en.wikipedia.org/wiki/Design\)](http://en.wikipedia.org/wiki/Design)). It involves defining the hardware and software architecture, specifying performance and security parameters, designing data storage containers and constraints, choosing the IDE and programming language, and indicating strategies to deal with issues such as exception handling, resource management and interface connectivity. This is also the stage at which user interface design is addressed, including issues relating to navigation and accessibility. The output of this stage is one or more design specifications, which are used in the next stage of implementation.
3. **Implementation:** This step consists of actually constructing the product as per the design specification(s) developed in the previous step. Typically, this step is performed by a development team consisting of programmers, interface designers and other specialists, using tools such as compilers, debuggers, interpreters and media editors. The output of this step is one or more product components, built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirements. For projects involving a large team, version control is recommended to track changes to the code tree and revert to previous snapshots in case of problems.
4. **Testing:** In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error-free and fully meet the requirements outlined in the first step. An independent quality assurance team defines "test cases" to evaluate whether the product fully or partially satisfies the requirements outlined in the first step. Three types of testing typically take place: unit testing of individual code modules; system testing of the integrated product; and acceptance testing, formally conducted by or on behalf of the customer. Defects, if found, are logged and feedback provided to the implementation team to enable correction. This is also the stage at which product documentation, such as a user manual, is prepared, reviewed and published.

5. **Installation:** This step occurs once the product has been tested and certified as fit for use, and involves preparing the system or product for installation and use at the customer site. Delivery may take place via the Internet or physical media, and the deliverable is typically tagged with a formal revision number to facilitate updates at a later date.
6. **Maintenance:** This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Typically, every change made to the product during the maintenance cycle is recorded and a new product release (called a "maintenance release" and exhibiting an updated revision number) is performed to enable the customer to gain the benefit of the update.

Advantages

The waterfall model, as described above, offers numerous advantages for software developers. First, the staged development cycle enforces discipline: every phase has a defined start and end point, and progress can be conclusively identified (through the use of milestones) by both vendor and client. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage, or of customer expectations not being met.

Getting the requirements and design out of the way first also improves quality; it's much easier to catch and correct possible flaws at the design stage than at the testing stage, after all the components have been integrated and tracking down specific errors is more complex. Finally, because the first two phases end in the production of a formal specification, the waterfall model can aid efficient knowledge transfer when team members are dispersed in different locations.

Criticisms



([article&siteId=&rsid=cbsitechrepublicsite&sl=&sc=us&assetguid=&q=&cval=177508;1064&ttag=&bhid=\)](https://adclick.g.doubleclick.net/pcs/click%253F%253DAKAOjstRVi1nNhQsWrmLIHcIEp9Wk0TbIpxNbm9dfQFuF_E_ArHUW5kACaCz6E_qh-Fo1o_wSGO0kJqUCNWFxD1dcQ_LMz2EATyfKwjJa017WbStuA04OX-7R663tWyNT2Qv-WGKwnctXExfZL9oXdeVPJb41UDJJRloP0OTA%2526sai%253DAMfl-YSKy7Xv4oEWE9I8ZV1zGVaBzAmwTau3nFbF3uq8a1l1JpZuRBC2jq-gOZQSx62NSSFIRIdPqISMtGw61zdJ_hZsKzOgGF0dXaPw3bdD7B7gm_pP59WebX1kIUyi4dckq-k7_w%2526sig%253DCg0ArKJSzAxCqBXiv70rEAE%2526urlfix%253D1%2526adurl%253Dhttps://cbsInk.cbsileads.edition=en&ursuid=&devicetype=desktop&pagetype=&assettitle=&assettype=&topicguid=&viewguid=97e965e8-c9ee-4908-8b1c-206f1700a097&docid=177508&promo=1064&ftag_cd=TRE-00-10aaa4e&spotname=dfp-in-article&destUrl=https%253A%252F%252Fwww.techrepublic.com%252Fresource-library%252Fwhitepapers%252Fadministrator-s-guide-to-disaster-planning-and-recovery-3rd-edition-pdf%252F%253Fpromo%253D1064%2526ftag%253DTRE-00-10aaa4e%2526cval%253Ddfp-in-article&ctag=medc-dfp-in-</p>
</div>
<div data-bbox=)

Administrator's Guide to Disaster Planning and Recovery, 3rd Edition - PDF

(<https://adclick.g.doubleclick.net/pcs/click%253F%253DAKAOjstRVi1nNhQsWrmLIH>

IT managers must make disaster planning a top priority if they are to prevent

eBooks (<https://www.techrepublic.com/resource-library/content-type/ebooks/>) provided by TechRepublic Premium

(<https://www.techrepublic.com/resource-library/company/techrepublic-premium/>)

Despite the seemingly obvious advantages, the waterfall model has come in for a fair share of criticism in recent times. The most prominent criticism revolves around the fact that very often, customers don't really know what they want up-front; rather, what they want emerges out of repeated two-way interactions over the course of the project. In this situation, the waterfall model, with its emphasis on up-front requirements capture and design, is seen as somewhat unrealistic and unsuitable for the vagaries of the real world. Further, given the uncertain nature of customer needs, estimating time and costs with any degree of accuracy (as the model suggests) is often extremely difficult. In general, therefore, the model is recommended for use only in projects which are relatively stable and where customer needs can be clearly identified at an early stage.

Another criticism revolves around the model's implicit assumption that designs can be feasibly translated into real products; this sometimes runs into roadblocks when developers actually begin implementation. Often, designs that look feasible on paper turn out to be expensive or difficult in practice, requiring a re-design and hence destroying the clear distinctions between phases of the traditional waterfall model. Some criticisms also center on the fact that the waterfall model implies a clear division of labor between, say, "designers", "programmers" and "testers"; in reality, such a division of labor in most software firms is neither realistic nor efficient.

Customer needs

While the model does have critics, it still remains useful for certain types of projects and can, when properly implemented, produce significant cost and time savings. Whether you should use it or not depends largely on how well you believe you understand your customer's needs, and how much volatility you expect in those needs as the project progresses. It's worth noting that for more volatile projects, other frameworks exist for thinking about project management, notably the so-called spiral model...but that's a story for another day!

WHITE PAPERS, WEBCASTS, AND DOWNLOADS

Okta + ICSynergy: Hybrid IT Got You Down? There's A Better Way with Identity
Webcasts from Okta

WATCH NOW

IT Help Desk Survival Guide, Third Edition (ISO file)
eBooks from TechRepublic Premium

DOWNLOAD NOW

TechRepublic Pro's New User Education Form
Tools & Templates from TechRepublic Premium

DOWNLOAD NOW

CXO spotlight: The risks and rewards of fast IT
Downloads from TechRepublic Premium

DOWNLOAD NOW

Track Project Requirements
Tools & Templates from TechRepublic Premium

DOWNLOAD NOW

EDITOR'S PICKS





Transgender employees in tech: Why this "progressive" industry has more work to do



Python is eating the world: How one developer's side project became the hottest programming language on the planet



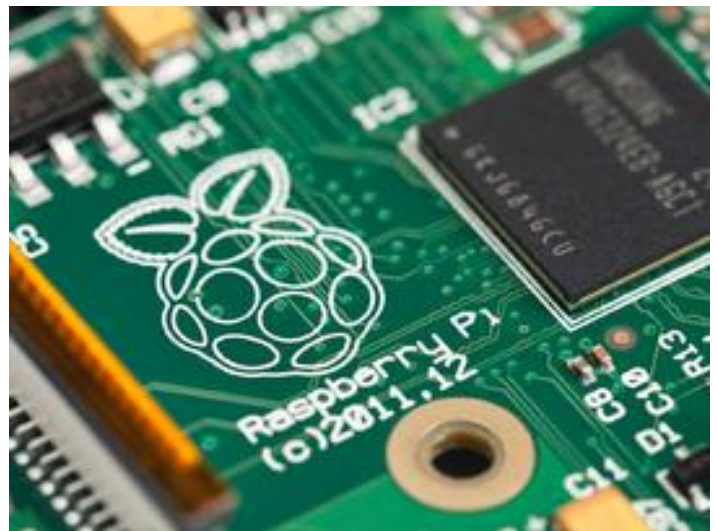
How iRobot used data science, cloud, and DevOps to design its next-gen smart home robots



Beyond the PC: Lenovo's ambitious plan for the future of computing



Straight up: How the Kentucky bourbon industry is going high tech



Inside the Raspberry Pi: The story of the \$35 computer that changed the world

- DEVELOPER
 - MOBILITY
 - TECH & WORK
 - CXO
 - SOFTWARE
 - HARDWARE
 - SMART PERSONS GUIDES
- DEVELOPER ON ZDNET 

Recommended

Promoted Links by Taboola

Wells Fargo Propel® Card

Wells Fargo

10 Credit Cards That Can't Be Beat In 2020

NerdWallet

Stop Wasting Money - This App Finds Every Promo Code on the Internet

Honey

Python programming language: A cheat sheet

Hyundai Motor and Uber release full-scale air taxi model at CES

WSL 2: where is it, and where is it going?

SHOW COMMENTS

EDITOR'S PICKS

Transgender employees in tech: Why this "progressive" industry has more work to do



Python is eating the world: How one developer's side project became the hottest programming language on the planet



How iRobot used data science, cloud, and DevOps to design its next-gen smart home robots



Beyond the PC: Lenovo's ambitious plan for the future of computing

WHITE PAPERS, WEBCASTS, AND DOWNLOADS

Quick glossary: Software-defined networking
Tools & Templates From [TechRepublic Premium](#)

[DOWNLOAD NOW](#)

IT Security Survival Guide, Second Edition

eBooks From [TechRepublic Premium](#)

[DOWNLOAD NOW](#)

Desktop Upgrades Pack

eBooks From [TechRepublic Premium](#)

[DOWNLOAD NOW](#)

Use our interview questions form to track and rate candidates

Tools & Templates From [TechRepublic Premium](#)

[DOWNLOAD NOW](#)