**One of the software project methodologies**

**---Waterfall**

By Lianyin Li

Waterfall is one of Software project methodologies, which is used for years. It contains such phases: requirements gathering and analysis, design, coding/implementation, testing, deliver and maintenance (Figure-1). Waterfall just like its physical substantiality that goes from top to bottom, and in software world, waterfall is linear and all phases take place in order. Each phase is completed fully then go to next phase.



Figure-1 waterfall model (refer from: <http://en.wikipedia.org/wiki/File:Waterfall_model.svg>)

Waterfall is working for years when it applies to small projects which last short period of time and with stable requirements.

Waterfall has a lot of advantages.

1. It’s simple and clear. It contains few phases and no increment and iteration without complex process.
2. It’s defined. Requirements are defined; each phases start and end time are defined. Everybody’s work is defined.
3. It’s easy to be managed and be tracked. Each phase there is a specified start time and end time. And defined works for each one well. And the requirements are stable during the whole project process. These help project manager or other people who involved in project manage and track their work.
4. It applies to small project well.
5. Requirements, analysis, design and so on all are documented well.

On the other hand, waterfall has some disadvantages, too.

1. It doesn’t suitable for large projects. Usually a large project contains tons of information and it’s impossible to collect all information in short time or reasonable time for that project. And for the fallible memory sake of people, is it easy to recognize what have done before months before even years before. I double it. If use waterfall in a large project that last years, maybe the spokesman in the customer side has been changed, do them remember what do they want when the project delivery after years? On the same reason for developers, it makes tough to recall what code they wrote months before or years when they try to fix some defects. And it put the project into high risk to be fail because of there are potential defects in coding phase until they are detected in testing phase even more critical in delivery phase.
2. It might be not appreciated by customers. Since customers involved less after requirements gathering and analysis phase. They always see/use the product after delivery phase. Before that, they have no idea what the earth the product look like or how does it meet with their requirements. Maybe it’s not the one customer real want.
3. It might make project costly. Do you see a real waterfall goes down to up? Of course no. So does waterfall in the software world. If something wrong found about requirement or design in coding phase, it’s impossible to go back to related phase. If some defects found in delivery, it could not be go back to coding and fix them. This might lead disaster to the project. The project might fail or re-do.
4. It might be not high efficient enough. There are two reason. One is all phases are in order, the next phase takes place only after the previous phase is completely totally. If the previous phase delay, after that all phases would delay. For example, if during the requirement gathering, if project team and customers could not stay on the same page. The requirements gathering would stall. Next phase could not be going. And other reason, customers sometime don’t know what exactly they want or could not press exact meaning in words. It makes difficult and take long time to gather all requirements correctly and efficiently.
5. It puts most stress on the people who gather requirements and analysis. The first phase is so important. It likes the base of the house. Once the base is built wrong or not strong enough, the house will crash someday.

In short words, which methodology should be chosen depend on various aspects: project size, complexity, and budget, how long the project would be going on. And how clear about the customers’ requirements. Waterfall has so many shortcomings, and there are methodologies have remedied these shortcomings, such as Spiral and agile software development. However, for a short period of time and small size that doesn’t involved so much people and requirements and customers doesn’t mind only last minute to see the product, waterfall is good choice.

Reference:

1. <http://en.wikipedia.org/wiki/Waterfall_model/>
2. <http://www.tutorialspoint.com/management_concepts/project_management_methodologies.htm>
3. [Melonfire](http://www.techrepublic.com/search/?a=contributor+melonfire) (2006) Understanding the pros and cons of the Waterfall Model of software development from

[www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/](http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/)

1. Mary Lotz (2013) Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?from

[www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology/](http://www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology/)

1. Jean Scheid, Michele McDonough(2012) Project Management Methodologies: How Do They Compare?

[www.brighthubpm.com/methods-strategies/67087-project-management-methodologies-how-do-they-compare/](http://www.brighthubpm.com/methods-strategies/67087-project-management-methodologies-how-do-they-compare/)

1. <http://pm.stackexchange.com/questions/389/when-to-use-watrefall-when-to-use-scrum/>