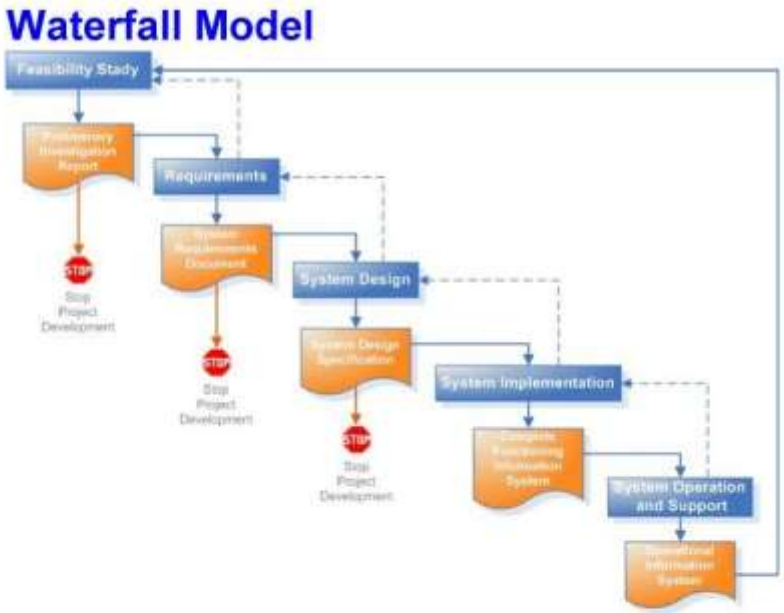
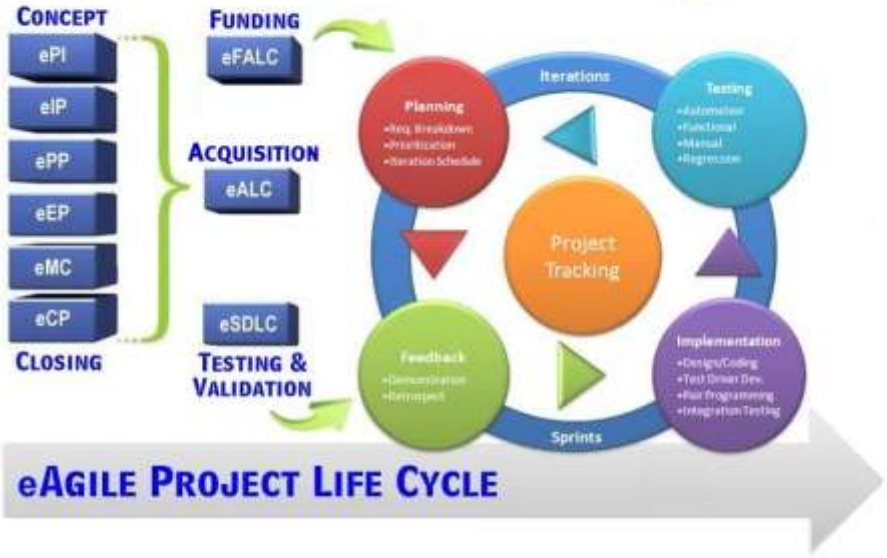
2012

Waterfall vs. Agile Methodology





Mike McCormick

MPCS, Inc.

Revised Edition 8/9/2012

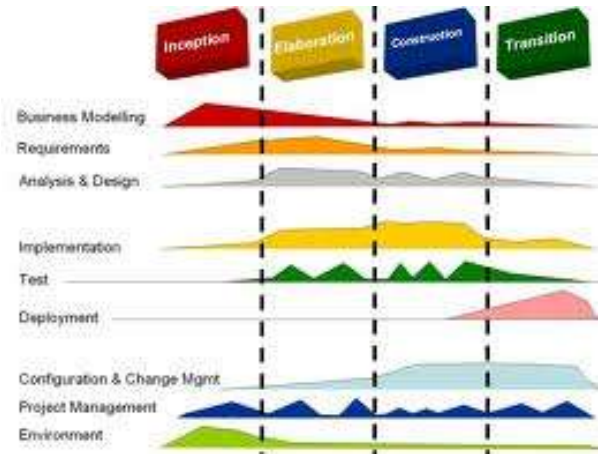
Waterfall vs. Agile Methodology

Contents

Waterfall vs. Agile Methodology

This article is a comparative analysis of waterfall model versus the agile model of software development. Checking out this analysis may help you in choosing which model is conducive for your own software development project.

Software development, like any other business process, has certain targeted goals which must be achieved in a fixed time frame. There are various strategies for achieving these software development goals. Two of the most popular software development models are '*Waterfall*' and '*Agile*'.



Waterfall vs. Agile Model Comparison

Waterfall model in software engineering was formally introduced as an idea, through a paper published by Winston Royce in 1970. However, it's ironic that he himself had introduced it as an example of a flawed software development method, that's vulnerable due to its many shortcomings. Nevertheless, every methodology has takers and this model has been successfully implemented by many software companies.

Waterfall model philosophy was inherited from the hardware manufacture strategies and construction strategies that were in practice during the 1970s. That's why; it has a very structured approach to software development.

On the other hand, the agile model of software development evolved in the 1990s, when developers decided to break away from traditional structured, segmented, bureaucratic approaches to software development and moved towards more flexible development styles. The 'Agile' or 'Lightweight' methods as they were called were formally defined in a research paper by Edmonds in 1974. Some of the most prominent and popular agile methods of software development, that subsequently evolved, are 'Scrum' in 1995, 'Crystal Clear', 'Extreme Programming' in 1996, 'Adaptive Software Development', 'Dynamic Systems Development Method' in 1995 and 'Feature Driven Development'. In 2001, a group of pioneers in agile software development came together and declared the 'Agile Manifesto', which is a set of canonical rules of sorts, for agile software development methods.

Conceptual Difference

Waterfall model, as the name itself signifies, is a sequential process of software development. Just like in a waterfall, the water progressively falls from one altitude to the lower, in a similar way, the production cycle progresses sequentially, from one stage to the other.

The phases of software development, in this model, are as follows: requirement specification, conception, analysis, design, coding, testing & debugging, installation and finally, maintenance. In this sequentially structured approach, the development team goes ahead to the next stage of development, only after the previous stage is fully accomplished. Software development companies, adopting this model, spend a considerable amount of time in each stage of development, till all doubts are cleared and all requirements are met. The belief that drives this kind of software development model is that considerable time spent in initial design effort corrects bugs in advance. Once the design stage is over, it's implemented exactly in the coding stage, with no changes later. Often the analysis, design and coding teams are separated and work on small parts in the whole developmental process. Emphasis is placed on documentation of every software development stage.

Waterfall vs. Agile Methodology

Now let's look at the agile software development method. Compared to the 'set-in-stone' approach of waterfall development models, the agile breed of models, focus on 'agility' and 'adaptability' in development. Instead of one time-consuming and rigid development schedule, agile models involve multiple iterative development schedules that seek to improve the output with every iteration. Each iteration goes through all the steps of design, coding and testing. The design is not set in stone and is kept open to last minute changes due to iterative implementation. The team structure is cross functional, closely knit and self-organizing. The design idea is never totally frozen or set in stone, but it's allowed to evolve as new ideas come in with each release. Less importance is given to documentation and more to speed of delivering a working program. Customers may be provided demonstrations at the end of each iteration, and their feedback may determine the next course of changes in the next iteration. The iterative cycle continues till the customer is delivered with a product which exactly meets his expectations.

Efficiency

In the ongoing comparison, let's see how these two ideologies compare with respect to development efficiency. Efficiency is decided by the quality of ultimate software product, the number of bugs and the development time consumed. Through my own research into the working of both these models, I found the agile models to be more efficient than the waterfall model, due to their adaptability to the real world. The '*One Phase*' and '*Rigid*' development cycle, makes it difficult to make last minute changes in requirements or design. While the agile methods, due to their iterative and adaptable nature, can incorporate changes and release a product, in lesser time. Of course, agile models are not perfect either, but they are certainly more widely applicable than the waterfall model. Of course, the expertise, skill set, attitude and ability of team members working in the project are a prime factor which affects efficiency. Be it agile or the waterfall model, that's adopted communication within the team members and with the customer, goal setting and better planning contributes to improving efficiency.

Suitability

The waterfall model is suited for development of programs that are already stable. That is, their design doesn't need a major makeover. In situations where the designers of a software program can accurately predict the flaws that may arise, in advance, waterfall model is the appropriate choice. Despite all its flaws, its design is easier to manage and the development costs can be ascertained beforehand. It is a natural choice when the customer has provided a clear list of requirements, which are not likely to be modified. On the other hand, when the customer is not clear about his requirements or expectations from the end product, adopting Agile model makes sense. The experience of the team members in handling the specific type of project should also be taken into consideration. If the developers are experienced enough at handling that project, then Agile approach is a better option.

Another point of consideration is the time frame within which the project is expected to be finished. When the time frame is long enough, choosing the Waterfall route is possible, while rapid delivery projects are best handled in the “Agile” way. Cost of the project is another point of consideration, which may influence your choice.

Agile models are applicable in every area of software development. It depends a lot more on the team effort of above average programmers, than relying on a few expert programmers. It's best suited for web based applications where its iterative nature helps in incorporating and correcting the various bugs that arise over time. Choose a model that best suits the experience of your team.

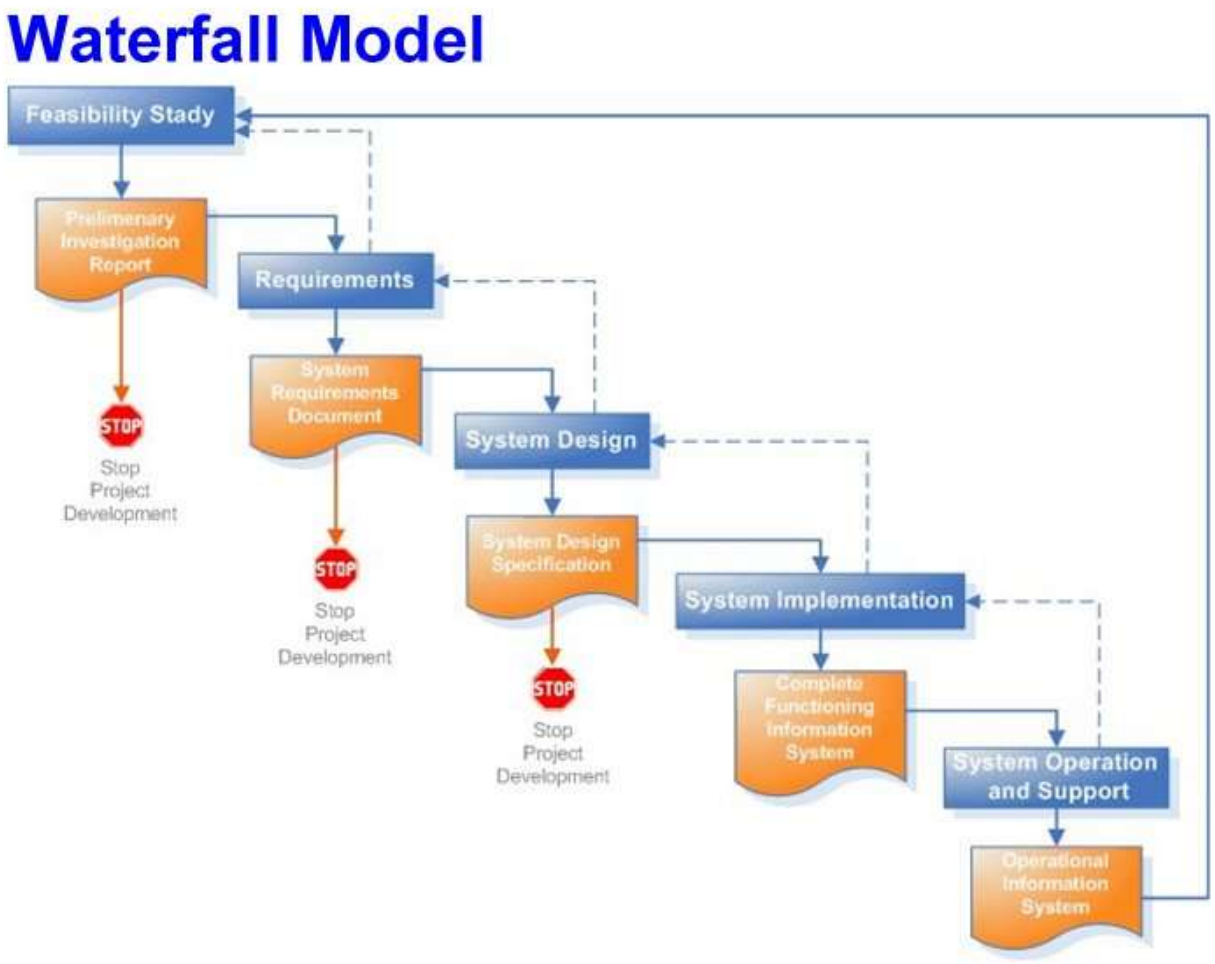
What is ideally needed is a model, which combines the accountability and predictability of a waterfall model, with the agility and adaptability of the agile model. That is, an effective balance between the two ideologies can create a more efficient software development model.

Waterfall vs. Agile Methodology

Waterfall Model Diagram

Since the time it was first published by Winston W. Royce in 1970, the waterfall model has been used widely in the field of software development. In the flow chart below, the waterfall model is illustrated and explained.

Well-designed objects have blueprints or plans. Once the design or structure is planned out, its creation is a simpler and more effective process. The same lies with programming. Programs can have algorithms or flowcharts, which plan out the steps and functions that need to be performed. But large-scale software and computer applications require a base programming paradigm or model. In the software development process cycle, programming models are used to plan the various stages of developing an application. One such model is the waterfall model.



Explanation of the Waterfall Model

Let us now take a look at the different phases of the waterfall model. One important aspect that is worth mentioning is that this model is designed such that until the preceding phase is complete, you cannot move on to the next phase of development. Progress flows in a downward fashion, similar to the way rushing water, from a height, flows downwards; hence the name "waterfall" was conferred onto this programming model.

Phase I: Requirements

The first phase involves understanding what you need to design and what is its function, purpose etc. Unless you know what you want to design, you cannot proceed with the project. Even a small code such as adding two integer numbers, needs to be written with the output in mind. Here, in this stage, the requirements which the software is going to satisfy are listed and detailed. These requirements are then presented to the team of programmers. If this phase is completed successfully, it ensures a smooth working of the remaining phases, as the programmer is not burdened to make changes at later stages because of changes in requirements.

Phase II: Analysis

As per the requirements, the software and hardware needed for the proper completion of the project is analyzed in this phase. Right from deciding which computer language should be used for designing the software, to the database system that can be used for the smooth functioning of the software, such features are decided at this stage.

Phase III: Design

The algorithm or flowchart of the program or the software code to be written in the next stage, is created now. It is a very important stage, which relies on the previous two stages for its proper implementation. The proper design at this stage ensures an execution in the next stage. If during the design phase, it is noticed that there are some more requirements for designing the code, the analysis phase is revisited and the design phase is carried out according to the new set of resources.

Phase IV: Coding

Based on the algorithm or flowchart designed, the actual coding of the software is carried out. This is the stage where the idea and flowchart of the application is physically created or materialized. A proper execution of the previous stages ensures a smooth and easier implementation of this stage.

Phase V: Testing

With the coding of the application complete, the testing of the written code now comes into scene. Testing checks for any flaws in the designed software and if the software has been designed as per the listed specifications. A proper execution of this stage ensures that the client interested in the created software, will be satisfied with the finished product. If there are any flaws, the software development process must step back to the design phase. In the design phase, changes are implemented and then the succeeding stages of coding and testing are again carried out.

Phase VI: Acceptance

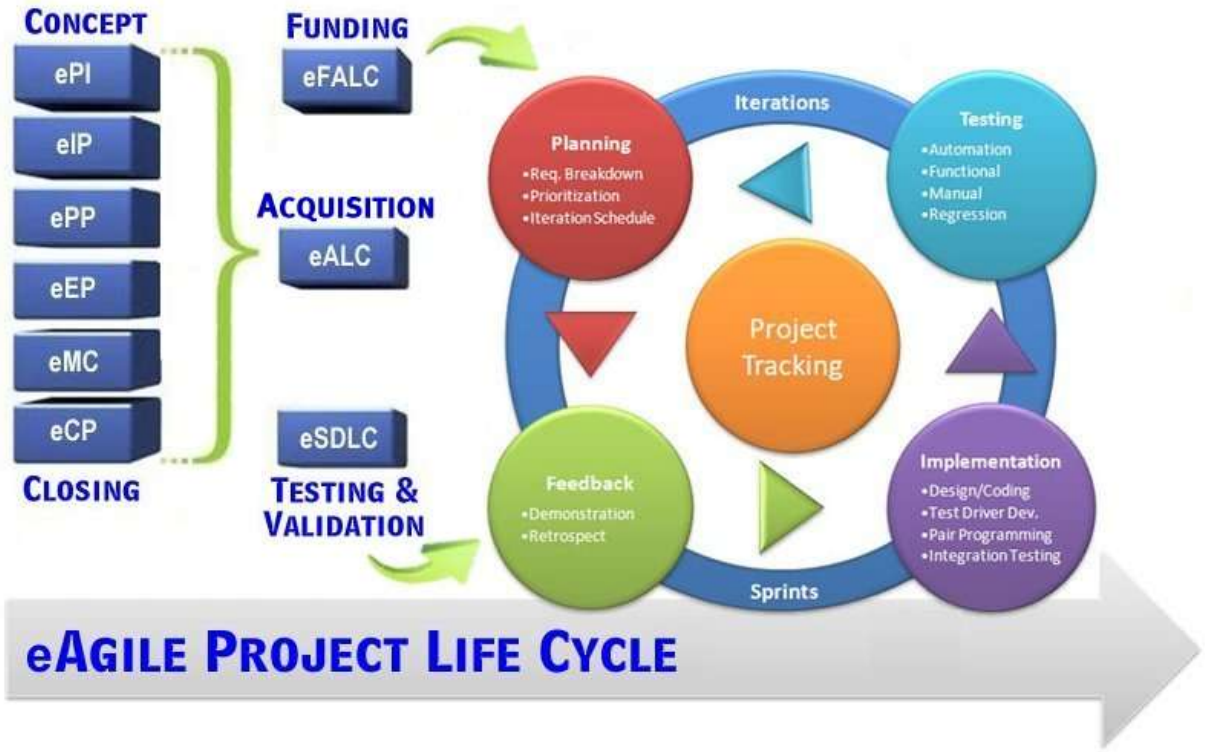
This is the last stage of the software development in the waterfall model. A proper execution of all the preceding stages ensures an application as per the provided requirements and most importantly, it ensures a satisfied client. However, at this stage, you may need to provide the client with some support regarding the software you have developed. If the client demands further enhancements to be made to the existing software, then the development process must begin anew, right from the first phase, i.e., requirements.

The waterfall model continues to remain one of the most commonly used methodologies. No doubt, new models have been used, but the widespread use of this model is the reason why it is studied in various software management subjects. With the above diagram in hand, you will not have much difficulty in understanding the process of software development. This is not only one of the simplest software process models for application development, but it is also known for its ease of implementation in the field of software development.

Waterfall vs. Agile Methodology

Agile Model

One of the models used in software development is the agile model and it is one of the latest models to be introduced in the software development industry. Scroll down to know more about this model along with the advantages and disadvantages of this model as well. In this model the “e” represents – enterprise.



Software development uses a planned and structured process to develop software products. The term 'software development', is the process, which involves writing and maintaining the code used to develop software. However, often the process of software development includes the idea of the software, designing the software through to the final development of the software, which is then handed over to the client. There are different models, which have been used to develop software. They include the waterfall model, spiral model, prototype model, incremental model, agile model, etc.

What is Agile Model?

The term agile stands for 'moving quickly'. This software development methodology is based on iterative and incremental model of software development. The requirements and the solutions are a product of collaboration between self organizing and cross functional teams. It is a lightweight software development model, which developed in the 1990s'. This model was developed against the background of the heavyweight models, which were famous for being heavily regulated, regimented, micromanaged.

The most important principle in this model is customer satisfaction by giving rapid and continuous delivery of small and useful software. The delivery of the software happens at regular intervals as opposed to after a number of months, which is the case with the waterfall model. The measure of progress used in agile modeling is the different working models delivered to the client. Since the software is developed in small batches, changes can easily be introduced into the software product. There is a lot of scope for cooperation between the business people and the developers, as the requirements keep coming from the business people at regular intervals. There is a lot of emphasis laid on technical excellence and good

design of the software. The software development team has to adapt regularly to the changing circumstances.

Agile modeling is a methodology, which makes use of practice for modeling and documentation of software based systems. Traditional modeling methods in software development projects have given way to these practices which are applied in a more flexible manner. It supplements the different agile methodologies like extreme programming, agile unified process and scrum models.

Advantages and Disadvantages

The most important advantages of this model are the ability to respond to the changing requirements of the project. This ensures that the efforts of the development team are not wasted, which is often the case with the other methodologies. The changes are integrated immediately, which saves trouble later. There is no guesswork between the development team and the customer, as there is face to face communication and continuous inputs from the client. The documents are to the point, which no leaves any space for ambiguity. The culmination of this is that high quality software is delivered to the client in the shortest period of time and leaves the customer satisfied.

If the projects are smaller projects, then using the agile model is certainly profitable, but if it is a large project, then it becomes difficult to judge the efforts and the time required for the project in the software development life cycle. Since the requirements are ever-changing, there is hardly any emphasis, which is laid on designing and documentation. Therefore, chances of the project going off the track easily are much more. The added problem is if the customer representative is not sure, then the project going off track increase manifold. Only senior developers are in a better position to take the decisions necessary for the agile type of development, which leaves hardly any place for newbie programmers, until it is combined with the senior's resources.

Agile Model in Software Testing

It is not only used in software development, but also for software testing. Agile model testing is carried out from the perspective of the end user. There is no emphasis, which is laid on the rigid testing procedures, but the focus is rather on conducting the tests iteratively on the newly developed software component, as well as regression tests are carried out on the entire software to check if any new bugs were introduced into the software. In the agile testing model, the focus shifts from 'testers as quality watchdog' to 'the entire team quality watchdog'.

As the name of the testing methodology suggests, the testers have to adapt themselves to rapid development cycles and make the required changes to the test suite. In this software testing type, the aim is to test from the perspective of the customer as early as possible in the development process. Because the testers are involved early on in the entire process of software development, they give the necessary information, feedback and suggestions to the development team, rather than after the development have come to the final stages.

Agile model has given the software development process an effective and practice based methodology. Therefore, the principle 'maximize stakeholder value' can actually be put into practice, leaving the customer satisfied and happy.

About the Author

Michael McCormick founder of MPCS, Inc and Management Professional with 35 years of experience managing over $4 billion in projects for both the Commercial and Federal Government sectors and is a well-known project management (PM) author, consultant, and authority on the subjects of Construction Management (CM), Facility Management (FM), Business Process Management (BPM), Project Management Office (PMO) and Project Portfolio Management (PPM), Risk Management (RM), software development and technology integration. MPCS Website: www.mccormickpcs.com