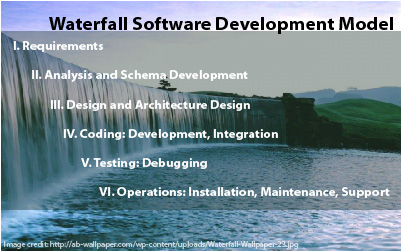
Alexander Ramsey  
UMUC SWEN 603

The Waterfall Software Development Model

In the 1970s, the field of Software Development was beginning to bloom and with it came the challenge of establishing a way to manage the people and planning behind a successful software implementation. Waterfall as a model was first described by Dr. Winston Royce in a paper titled, “Managing the Development of Large Software Systems.” In this paper, Dr. Royce described his approach to the management of software development. The manufacturing industry had for the most part utilized a model similar to that proposed by Dr. Royce (Royce, 1970). Waterfall models this industry due to the rigidity of manufacturing and the importance of good design before mass production. Waterfall represents very important core management functions in production but has recently become a model of the past when considering the demands of a dynamic and rapidly changing marketplace. Disruption in industry caused by this rapid change has caused other methods of development to take the focus away from Waterfall and deem it a thing of the past. Dr. Royce describes the model as having six phases, shown below:



During the Requirements phase, the software team works with the client to establish exactly what is to be developed. We can relate this phase of development to an initial consultation, where the client shares the business or industry objective behind the implementation of the software solution. This phase is often represented in physical form by a Product Requirements document. The Product Requirements document will describe distinctly what the software program will do for the client. Note that this document does not describe exactly how it will be done.

The Analysis phase involves unwinding information gathered in the Requirements phase to determine how a technical implementation can occur. Actions taken during this phase may include the development of a Database Schema for data warehousing, or the development of Business Models that will represent the core Business Logic of the application. With this, may come further insight into further requirements that must be considered to fulfill the initial objective. At this point, the development team can return to the Requirements phase to make adjustments and account for additional or reduced functionality.

Following the Analysis phase is the Architecting phase, where the pieces of the puzzle all fit together to form a system design. This design will involve a larger overview of the necessary systems and components that will fulfill the requirements of the application. This can be a very important step in the process because the core structural decisions must be made before the development of the application is established. Decisions made during this phase can be costly later due to the way that each piece must be designed to fit together and be somewhat rigid.

The next step of the process is where Software Methodologies as a study are aiming to improve. The Coding step is where the rubber meets the road for the development of the analyzed and architected requirements. The effectiveness and efficiency of this step must be well estimated to ensure that client expectations are met within budget and timeframe. Many unforeseen circumstances can arise when considering the technical implementation of the requirements. Waterfall can contribute to success of a project as the design process can help a less experienced software development team to understand more of the structural elements before the development phase begins.

The Testing phase of the model also represents an important area where the requirements and analysis may change due the modification or discovery of additional resources pertaining to the development of the software. Debugging and testing is predictable for applications where the initial design phase is effective enough to determine exactly what the requirements are of the client and for the successful technical implementation of the requirements. So, if during the testing phase there are gaps in the desired functionality, there may need to be a full review of the project starting again with the requirements phase. However, in scenarios where initial design phases are performed in a systematic and effective manner, the testing phase may go very smoothly than if a project did not receive as much structure in the initial phases of development.

This leads into the final phase of the development lifecycle presented by Waterfall, that implementation and adoption determine the ultimate success of a project. This phase allows for room for the development of a software program to further develop over time while the client supplies feedback and important information regarding the functionality requirements as they are discovered.

The Waterfall Model addresses many concerns in the development of a software program that may require a strict process, where the complexity of the end product may be high and the requirements be more specific. In fact, the Waterfall model exists because of the increasing complexity and reliance upon software during the advancement of technology to adopt software programs in large scale enterprises and military intelligence resource pools. Waterfall represents the simplest of models due to the fact that it is based on core management principles we can relate to in the manufacturing industry. Although it may be considered to be out of date, it provides a clearly outlined and relatable structure for the development of simpler software projects.

Royce, D. W. (1970). Managing the Development of Large Software Systems. Retrieved from http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf