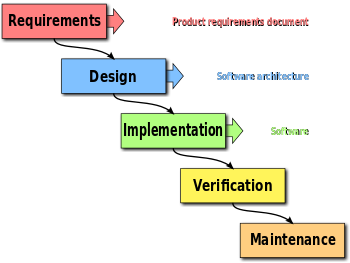
Ashley Kodya

08/2015

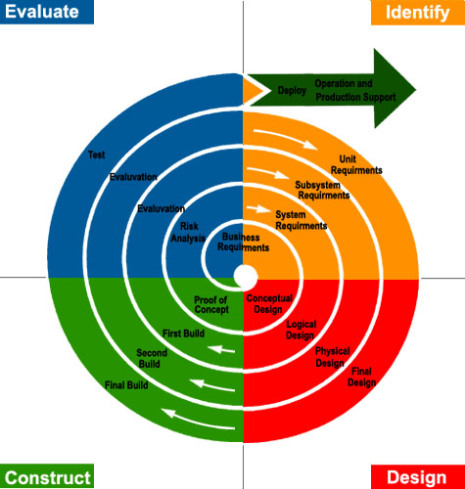
Network and Server Structures

Section 01

The two software development lifecycle models I chose to review were the Waterfall model and the Spiral model. The two mostly opposite but share some similarities.



The waterfall model is your basic SGLC model and intended to be used for small budget and minuscule projects. The Waterfall model is linear and you basically just follow through with the steps in sequential order. The first step is Requirements where you define what your software needs to include and what the user needs to have access to in order to completely use the software. Next you design the software under the Design step. Then you implement the software, verify if working okay and maintenance work is down to finalize the design.



The Spiral design is a great tool for convoluted and high-risk software where there will need to be room to go back and improve as the development continues. The spiral model deals with long-term projects and the requirements are a bit more complex. The steps are: Evaluate, Identify, Construct, Design. This is called a spiral model because it is opposite of linear; you can jump through the different steps to complete your software design.

The Spiral design differs here because it is intended for where the waterfall deal with simple and low-risk software that is sequential and linear, the spiral method is for high-risk analysis that deals with many variables and changes. Actually in this sense, the model is quite opposite. In the spiral model, the developer can jump back and forth through the steps, as opposed to the waterfall model where once you go through one step you move onto the next one and do not go back. The spiral model is ideal if it is known before going into development that there will be a large amount of testing and debugging that will cause the developer to go back to the design stage. Also the spiral model differs because it’s intended for large projects and the waterfall model should only be used for small-scale projects. Another huge difference with the two models is that in the waterfall model if you want to add further functionality you basically have to start from square one and rebuild a new design. With the spiral design, you can go through the steps as you wish, as it is what is intended.

As mentioned before, the two are pretty much polar opposites. Really, their only similarity is that they are both system development cycle models. The two models are seem like seamless designs to produce great software as long as you use them properly.