SDLC Comparison

By: Trenton May

Class: Networks and Server Structures

Instructor: Joseph Dalton

Conclusion

Both of these SDLC models are useful, both have their good sides and their bad sides and should be used appropriately according to their weaknesses and strengths, respectively. For the most part the Waterfall and Spiral models are polar opposites. While as the Spiral model produces better results it is too costly and should only be used for larger projects while as the Waterfall model is cheap and should be used for small projects only but also has higher risk. The Waterfall model also only visits a step only one time and then moves forward after a review, while as the Spiral model has multiple steps that it goes over many times to ensure the best quality and cost efficiency. These two models really don’t have any similarities due to the fact that they were built for two different scenarios and with that statement these models should be used appropriately. Choose wisely

Waterfall Model

* Simple and classic style of an SDLC
* Every phase is independent of each other. What this would mean is that the step that you are currently on must be completed before moving on
* This is ideal for small projects due to the simplicity of the architecture of the planning and the need to complete each step before moving on to the next
* Typically near the end of each phase there is a review of the work that has been done so that there can be quality assurance that it is ok to move on to the next step
* Unfortunately there is a higher level of risk and uncertainty with this model. Leaving it ideal for small projects and not ones that will cost the client or the company a large amount of money
* No actual software is produced until late in the waterfall model.

Spiral Model

* This model has a higher outlook and awareness for the amount of risk that goes into the project
* The spiral model is ideal for larger projects due to the fact that multiple prototypes are typically produced throughout the time that this model is used
* Early on in the project software is produced and is able to be used.
* The success of the project is determined by the risk analysis which happens multiple times throughout the project (at least every time a new prototype is created)
* Costs more money than some other SDLC’s which makes this mostly ideal to be used in larger projects exclusively.
* Multiple phases of design, analysis, development, and planning are carried out to ensure the best possible end results, which again make this ideal for large projects.