What is SDLC?

SDLC also known as software development life cycle is a process that will assist a software production team from the planning phase of product to the deployment phase of a product by allowing the team to have a detailed plan on how to develop, maintain or make whatever changes that are necessary according to the scope of the project. The life cycle will consist of six stages that will be used to improve the quality of the software and help with the overall development process.

The first stage would be to plan for the overall development. This includes gathering inputs from the clients, gathering survey data from the target audience. Risks and quality assurances should also be discussed and identified during this stage. The ideal outcome of this stage is to implement a project that will result in a quality product and comes with minimal risks.

The second stage will be defining the requirements of the products. The scope and the objectives of the project will be discussed in this stage. The team will be doing an analysis to define the requirements and will also be constantly reviewing the requirements with the client for approval. After the client has fully approved of all the stated product requirements to be designed and developed the team will create a document which will contain all the product requirements to be referenced to throughout the whole development cycle.

The third stage would be on designing the product’s architecture. The product’s architects will reference the document created in stage 2. The architecture type will be proposed and documented in another document the document will be reviewed by the team and the clients. The architecture would be based on a multitude of things ranging from risk assessment, how modular the design is, budgets and time constraints. The most efficient way would be put forth in the future stages of the development cycle.

The fourth stage would be building the product, the product would have to follow the guidelines of the document outlined in stage 2 and stage 3. The fifth stage is the testing stage of the product where all the modular sections or the whole product depending on the architecture would be tested.

After the testing stage it will be the last stage of the development life cycle. The product would be ready to be deployed to the market according to the client. The product might be changed at any time after this stage based on the feedback of the product.

One model would be the waterfall model. The waterfall model has its phases done in order. With the next phase only started when the previous phase has been finished. The advantages of the waterfall model is that it is simple and easy to understand, phases are also completed one at a time allowing members of the team to be able to be very clear on what stage of the development life cycle they are on. However, the disadvantages of the waterfall model is that when it is in the testing phase it is very hard to go back to previous stages to change something if the phases were not well documented. There will also be no working software until late in the cycle.

Another model would be the V model. The V model is almost the same as the waterfall model. The V model requirements would have to be very clear before the project starts because like the waterfall model it is very difficult to go back to previous stages.

There is also the agile model for development. The model takes on a different kind of approach from the previous two models. Where the previous two models have a predictive approach where the team would be working with detailed plans and have documents properly outlining the tasks that they are supposed to do however the adaptive approach that the agile model uses has no detailed planning and the future tasks is based on what features that the product requires. The advantages of using the agile model is that it’s a good model where the environments will change frequently. The whole model is very flexible and easy to manage. The model also requires little to almost no planning. However the disadvantage is that it requires a very good teamwork between the team members and it’s not suitable for handling complex dependencies. The model will also come with more risks to its sustainability and maintainability.