

SDLC Comparison

Rapid Application Development vs. Spiral Development

Rapid Application Development is similar to what I have used in my class projects. Developing an UX Plan which would describe what the main intent of the project is, describe some test users on each end of the spectrum (seasoned computer users and complete novices) and list required features that the client wants and ones that would be nice to have. After creating some mock-ups and running some nav-flow testing you're ready to start coding and testing your code. This Rapid Development approach works very well on small-scale projects and allows you to complete the project in a faster manner. This saves the client money and allows for the company you work for to do more projects. If you were to try this on a large-scale project you would run into more problems than you would want to deal with because of the complexity of the project. It would be better to have a more in depth model that has more hands on deck and more levels of creation and development.

The Spiral Development model looks to be one of the best models as far as covering all aspects of development and double-checking to make sure you have a product that is user friendly and works right out of the gate. The only problem with this model is that it is quite time consuming and if you have a budget to work within that needs to be considered. On large projects this would be ideal as there are many components and pieces that need to fit together perfectly at launch. There is a problem with the Spiral Development model that I would definitely adjust if I were running a project on a large or small scale and that is to start testing after the first prototype. If you just do one round of testing then you run the risk of having to go through another design phase to fix some issues that might come up during testing.

If there is one thing I have learned so far it's that even though I see that the steps are easy to follow when going through say a create a new account or find a certain spot test its that everyone isn't me. Yeah of course it's easy for me because I am the one sitting in front of the computer creating it, but when fresh eyes come up and look at it you want them to be able to scan and find what they are looking for fast. That is why testing is so crucial because you don't want to loose customers because it took them too long to find something so they went to another site. Those are issues that can be resolved with testing and that is the great thing about the Rapid Application Development Model.

So in conclusion I think that no matter how big or small your project is there are key elements that you need and if you have those then your project will run smoothly and stay on track. Planning, Communication and Testing are all keys to a good Development plan, at least to me. Obviously you are going to have a build, design and other phases to your model but the most important are planning, communication and testing because if you don't have these then you don't have a successful project and it will never get off the ground. From planning to development there needs to be accepted two communication between all levels so that everybody is on the same page and can see the project through to the end.