Landon Leigh

ITSC 3155 Final Exam

1. When you are talking about what software modeling is, you are talking about the entire picture in respect to software design. By this I mean everything from all the interfaces, methods, and the way that they interact with other software. Software modeling includes the diagrams (from the simplistic to the detailed) of what your software will encompass. You can map out the parts of your project to help you during the creation process. One useful tool that can be used for software modeling is UML. We need to use tools such as UML for software modeling because it helps us model and label the parts of the planned software. It allows us to look at the diagram and get a good understanding of what certain parts should accomplish and what types they should be. Some software modeling tools include UML, Lucidchart, Dia, Gliffy, Microsoft Visio, Enterprise Architect, and many more. You can use software modeling to realistically estimate the amount of work that is needed to be done per task by modeling out each specific part of the tasks. By doing this, you can plan out how long you believe it will take to complete each specific part and you can use all of the combined times to give you an estimate of how long each task should take you. This will help you determine if you are behind or ahead of your planned schedule to tell you if you need to work quicker or if you have more time to spend on certain parts.
2. The general idea behind the principles of Manifesto for agile software development is that the goal is to ensure that the software can be worked on at any point and that it is completed in quick and continuous fashion. The software needs to be delivered continuously delivered so that there will be feedback so that the software can be changed according to the needs of the customer and so that the product will be created in the best and most efficient way possible. One of the principles I have chosen is “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software”. This principle means that during the development process, the most important goal is to make sure that the customer is happy with the product receiving. This can be done by providing the software to the customer in stages as they get completed so that their feedback can be used to finish the project. The other principle that I have chosen is “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly”. This means that during development, the development team will go over and look at how everything is being completed and if it is not quick enough, they will make a way to speed it up, or if it is going in the wrong direction, they will correct it and focus on the goal. I believe that these 2 principles are very essential to the project development because they work together to ensure that the customer is getting what they want. That is the most important goal in the development is that the customer is getting the correct software that they are paying for. The two principles work off one another because when they get feedback, they can tune their behavior which allows them to deliver to the customer much more efficiently.
3. Agile software development is the preferred method for the development of software projects because it is far better in multiple ways. Agile development is much quicker, consistent, easier to update, and much more flexible than other methods such as the waterfall method. Scrum is a part of agile development that is much preferred because it allows the software to be delivered in increments known as sprints. These are typically every two or so weeks and it allows the product to be presented so that feedback can be given. This is beneficial so that they can add more and fine tune what they have. Agile is also preferred because with waterfall you must follow a liner path of steps and if something needs changed you must pretty much start anew. When it comes to agile however, you can go back and work on many different parts at different times which is very beneficial. Agile is also better for the customers side because it allows them to see that progress and how the software is being made. It allows them to determine if they are satisfied with what they will be receiving. Lastly, scrum is very useful because during development, when the development team is finished with their sprint, you can plan and update then next and following sprints as you go.
4. Software architecture is basically the organization and planning of the development process. Using the term software architecture puts more emphasis on how important the task is and makes it sound like it is more expensive and in depth. When talking about the architecture you sound like you are talking about the important stuff. In architecture there is application architecture and enterprise architecture. The main type for us is the application architecture which includes the code base and such. When talking about software architecture we are basically talking about the outside components (i.e. the external components) whereas when we are talking about software design we are talking more about the internal components. Software architecture is the broad picture of what we want the software to do while software design is about how we want to work towards what we want the software to do. Software architecture includes topics such as the language used and the design patterns and software design is more of how certain classes will be used to do certain tasks.

References

<https://sea.ucar.edu/best-practices/design>

<http://agilemanifesto.org/principles.html>

<https://martinfowler.com/agile.html>

<https://www.guru99.com/agile-vs-scrum.html>

<https://www.360logica.com/blog/why-agile-development-methodology-proves-to-be-a-better-choice-for-your-project/>

<https://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>

<https://www.youtube.com/watch?v=VjKYO6DP3fo>