COSC 310: Software Engineering

Assignment # 1

Drew Ingram: 16680076

Due: September 21, 2012

I did not have a textbook or reliable notes so I used this as a reference:

<http://www.ijcsi.org/papers/7-5-94-101.pdf>

1. Differences between the Spiral model and XP model:

--In XP, every small piece of the program is delivered one at a time to the client. The spiral model also requires much communication with the client, but only to assess the work that has been done, not to deliver the system piece by piece

Example, Facebook: XP may deliver the entire database first, then a personal profile page, then a search page, then a wall etc. Using a spiral model, these may be prototyped for approval, but not completed.

-- Though both require a great deal of communication with the client, XP requires that the client is constantly on site, helping with the project as it develops. In the Spiral model, on the other hand, the client is only contacted after certain iterations through the spiral model.

Facebook example: Using the spiral model, the customer would look at the various prototypes and give their opinion. In XP, they are always present and give constant feedback as the project goes along

--The spiral model is meant for larger projects than the XP model, hence the greater risk analysis. If XP is applied to a large project, it may be difficult to piece together the various small parts because there are so many.

--XP requires the tests to be written before the development, but in the spiral model, these are most likely done during the development stages.

Facebook example: In XP, before implementation of the friend finding feature, particular users may be hardcoded with particular data. A test of the friend finder might involve creating a list of users that should appear as suggested friends for one particular user. The team would work towards passing these tests for to complete the friend finding feature using an XP model. In a Spiral model, these would tests may be created after.

2) For the project, shall be using the V-shaped development course. This is mostly for its simplicity, but also because it is test-driven, and I have some project experience with test driven development where I developed an algorithm by looking at possible scenarios my program could encounter and choosing the best outcome. There will be less emphasis on planning and specification analysis because it is already very clear what the program is supposed to do.

5) My partner in both questions was Bryan Lennox. I have known him for over a year and worked on a group project with him before. I also talk to him often when I see him. For the first question, I was the pilot. I have slightly more experience programming in Java (the language we chose), so we both felt I was suited for this role in this question, which was the more complex of the two. It was similar to programming on my own, but I feel that having Bryan read over my shoulder made things go much more quickly. First he would occasionally catch my typos which would have caused bugs that may have been very difficult to detect. More importantly though, I found I was more focused on the task than I usually am because I often get distracted.

For the second question, I acted as the co-pilot. Because Bryan had not programmed in Java for a long time, I mostly reminded him of java syntax and gave him hints. He had a slightly different idea of how to approach the problem, and I decided it was best to make sure I let him to do it the way he wanted and provide him guidance, rather than just tell him how to do it.

I imagine working with somebody I did not know might be somewhat frustrating at first. I would need to get to know him and his coding style and I might be too cautious or humble as the co-pilot to give any useful advice. I feel that once I got to know this person, I would be able to produce much more code working with him than on my own.

6) The traditional models are designed to be strict, while the agile models are meant to be flexible.

The key points of traditional software models are:

Planning: In traditional management activities, the planning is done at the beginning. Since there is may be so much feedback from the clients in agile methods, plans might change and all the time spent on planning at the beginning could be wasted.

Scheduling: The team may have to update the schedule more often because of the more constant feedback from customers.

Monitoring: There is less monitoring done by the team because the customer is more involved in the process of making the system.

Quality control: There is less emphasis here because the customer is more involved in the project.