

Assignment 2

1. The budget and the development time are the biggest concerns of any software project. The time it takes to develop a project is valuable if it is short just as much as a project that has a fairly low price. However, both the budget and the development time depend on each other. Such that the longer it takes, the more is bound to cost. Therefore, I don't believe I think that one concern could be more valuable than the other. The complete functionality of the project will be determined by these two as well.
2. In the Agile method for software development, there are four main phases that occur in each and every iteration. They are: building requirements, designing, writing code, and testing. I believe that none of these four has such a big predictable pattern that would allow you to do it only once and not at every iteration, especially in the case that someone wanted to save time.
3. The Waterfall method for software development has four main phases that occur: design, develop, test, and maintenance. In essence, they are the same as the agile method. However, their order differs in a way that puts more emphasis into the design, this is widely used in projects where the organizing of the interface will determine the user friendliness.
4.
 - a. The user story is the navigation experience the user gets from each interaction with the software.
 - b. Blueskying: A fancier term that pretty much means brainstorming, although performed in a manner that brings better ideas.
 - c. The user story should be concise, agreeable in general understanding, and effectively descriptive of the software.
 - d. The user story should not be overwhelmingly long, containing unconventional language, and describe specific aspects of the software without context.
5. I agree with the saying "All assumptions are bad, and no assumption is a good assumption." as it relates to user stories. The only thing that an assumption creates is more uncertainty. It's better to have clearer and executable user stories that don't depend on any unknown variables.
6.
 - a. You can dress me up as a use case for a formal occasion: User story
 - b. The more of me there are, the clearer things become: Estimate
 - c. I help you capture EVERYTHING: User story
 - d. I help you get more from the customer: Estimate
 - e. In court, I'd be admissible as firsthand evidence: Blueskying
 - f. Some people say I'm arrogant, but really, I'm just about confidence: Observation
 - g. Everyone's involved when it comes to me: User story

7. Better than best-case means that the programmer will not ignore any possible variable that could affect his or her estimate.
8. I think that the best time to tell your customer that you won't be able to deliver it on time is right when you realize that. This is because transparency is far more valuable than reaching a point of no return where a conversation is difficult.
9. Branching in your software is good because in a scenario where there are more inner parts of the project to be developed, separating in groups could nurture the success.
10. I haven't personally used any build tools for any project as a computer science major, nonetheless, I am aware that the benefit comes in execution of high level software.