Sophia Prochnow
30 October 2019
CMSI 401

Assignment 2

1. The two main concerns of any software development project are “how much will it cost?” and “how long will it take?”. It is no surprise that the two main concerns of anyone who is attempting to build a software application is using the least amount of time and money as possible. Personally, I believe the question of time to be the most important aspect. Especially in bigger companies, money can flow without a lot of changes to the company. Time, however, is something that once gone, can never be recovered. While it is ideal to accomplish both quick and low cost projects, if more money needs to be used for the purpose of saving time, I believe that to be the correct choice. This relates to complete functionality because there have been many instances where cutting corners with both time and money have created errors in applications that could have been avoided. Complete functionality should be the absolute highest priority, well above time and money. Without complete functionality, it doesn’t matter how much money or time was saved, the project is a failure.

2. The four phases in each and every iteration of AGILE are:
    - Requirements- understanding what the specifics for this particular iteration are and how they need to be completed
    - Design- mapping out the way the current goals of the iteration look/will be created. It is important to map out all of the requirements of the task and what it will need to contain to be considered complete
    - Code- this is to take the design that was previously made and implement it
    - Test- it is essential to extensively test all code produced to ensure that it covers every requirement specifically stated

I think some of the requirements of the project can be done at the beginning of the project, but it MUST be revisited at each iteration to ensure that the requirements for the upcoming task are fulfilling every request from the user, especially requests that have been adding once the project has already begun.

3. The waterfall method has seven main phases: conception, initiation, analysis, design, construction, testing, and deployment. The phases are similar to the Agile method, but contain a lot more work before the construction of the application actually begins. It includes conception of the idea, which Agile assumes has already happened when beginning each iteration. Waterfall does not use iterations, so it must ensure that all bases are covered before construction begins. I think a lot of the methods in waterfall are more or less assumed in Agile, but since there are no iterations in waterfall, it is important to specify them during the process. I think the analysis step is very important and an Agile method could benefit from adding it to each iteration. It encourages a

higher level of thinking about the application before the coding starts and could save time in errors or incorrect functions.

4.
   - A user story defines the exact step by step process of how the user interacts with a certain application to reach the desired goal(s) while containing all of the specified requirements; there will likely be multiple for each application.
   - Blueskying is a technique used at the beginning of the project cycle where each person working on the tool essentially throws out every idea they have for the project, just to have a collaborative and brainstorming environment.
   - User stories should describe one thing that the software needs to do for the customer, should be written so the customer can understand it clearly, be written by the customer pertaining to their exact needs, and be no more than three sentences.
   - User stories should NOT be long essays, use unfamiliar technical terms, or mention specific technologies.
5. I think the first statement is true. Especially when dealing with another customers idea for the project, you should never make any assumption about how they would want something to be. All decisions should be cleared by the customer. I think the second statement depends on what would be considered a “big” user statement. I think for some applications, it could be difficult to create a user story no more than three sentences, but the user story should absolutely not be more than one paragraphs length.
6.
   - User story
   - User story
   - Blueskying
   - Role playing
   - Observation
   - Evidence
   - Estimate
   - Blueskying
7. A better than best case scenario is when a programmer does not consider all possible scenarios and makes general assumptions that lead to an incorrect estimate of the time it would take to complete the project. It typically does not take account of all of the user stories or the time it would take to test the application.
8. I believe that you should inform your customer that you cannot meet her delivery schedule the second you become aware of the fact that you cannot do so. It would be unfair to hold onto this knowledge and not make your client aware that the timeline has changed. This is, of course, easier said than done as it would be a very difficult conversation to have since the client will most likely be unhappy with that news. All you can do is explain to them what caused the delay and assure them that you will do everything in your power to prevent any more delays.

9. I think branching in your software development is a good idea. Say, for example, your application has many different functions that the user can call. It would make no sense to try and code each function at the exact same time, but instead go one by one ensuring that they each work. In this case, once you have completed and tested a function and know it is fully working, you can push it to the main branch. Then, in a new branch, you can start building the next function without jeopardizing any of the working functions.
10. I have used a React app build tool for my interaction design project, called from the command line. Using this tool, it creates a skeleton containing a main javascript page, CSS, and a test suite. This is the only build tool I have used.