Conner-Ryan Petersen

Lawrence Benitez

Aiden McDougald

Leo Dai

Evan Steinhoff

Senior Project Assignment 2

Problem 1:

Two major concerns when it comes to a software project is how much it will cost and how long it will take. Personally I think time is more important because having a short or long amount will impact the development of whether people will cut corners or will be able to flesh out their ideas. For complete functionality in terms of budget is to make sure the customer is not being billed more than agreed upon. In terms of time it is delivering the software on the agreed upon time.

Problem 2:

The five main stages in the Agile method for software development are brainstorming, design, development, quality assurance, and deployment. No, personally I think each step is essential in providing quality software. No, if I were to start on the development phase I would eventually have to go back to the design phase and then the brainstorm phase to get an idea of where I would want to take the project.

Problem 3:

The main stages in the Waterfall Model are Requirement Analysis, Product Design, System Design, Coding, Testing, and Maintenance. The main difference is that Agile requires iteration while waterfall doesn't. At the end of Waterfall is maintenance meaning you are still working on that project while in Agile at the end you move on to the next project. A situation where Agile could use a step of Waterfall would be launching a video game, but you still need to maintain it by releasing patches.

Problem 4:

Blueskying is getting a bunch of people in a room together and getting as many ideas as you can. A user story is the customer's experience with the software that the team is building. Four things user stories should do is describe one thing that the software needs to do for the user, written with language that the customer understands, written by the customer, and be short. The user stories should not be long, use technical terms to confuse the customer, or mention specific technologies. No Waterfall doesn't have user stories.

Problem 5:

I like what the book is saying, we should always aim to test our assumptions because the more input we get from the customer the better our product can be.

I agree with what the book is saying, a big user estimate can take way too long to complete.

Problem 6:

1. User story
2. Observation

3. Blueskying
4. Role Playing
5. Observation
6. Estimate
7. Blueskying

For number 2 I think observation could also help with the user story to clear things up as much as possible. I agree with number 3 and 4 Observations would also help capture more information.

Number 7:

Everything is gonna go perfectly! Saying you can finish something by the earliest time when you don’t account for things like sleep or going out or gaming.

Problem 8:

The best time to tell your customer that you won’t be able to meet their schedule is before Milestone 1.0. This is the best time because we are addressing the problem before it gets worse. If we wait for Milston 1.0 we risk pissing off the customer more than having the conversation sooner. The conversation would be difficult, a way we can make it less difficult is having a new schedule prepared so the customer knows we have a plan already.

Problem 9:

Branching your software configuration is always good in development because it gives the developers a chance to test out new implementations without potentially harming the main branch. A scenario would be releasing a test server patch for a game. You would release a secondary branch of the game with the implemented changes so the players can give feedback before releasing it to the main branch.

Problem 10:

We haven’t used a build tool in our current development for Senior Project, but I have used Unity’s build tool to compile my Game Dev project. It was super easy to use. It compiled everything that I made and it ran smoothly.