Exercise 3: Software Process Models

Kendra Birringer (1229372) Nader Cacace (1208115)
Steffen Hanzlik (1207417) Marco Peluso (1228849)
Svetozar Stojanovic (1262287)

November 9th 2019

1 Case 1

1.1 a)

Another company built the existing software project, so we have no idea of the existing software project. Since the other software company doesn't exist anymore, so we are not able to contact them.

Requirements change very often. It is questionable, if we know, what the customer really wants.

1.2 b)

Try to search for similar projects and compare the "work load". Speak with the customer and try to get all information that we need to work on the project. With these informations we try to estimate the best possible price.

Maybe see the other software to get a feeling for our task.

Cheap starting price and for additional features the price will increase.

1.3 c)

Incremental, because we can start with less analysis of the customer requirements at the beginning (it's possible, that we don't have all informations).

We can work very close with the customer and if the customer changed his idea, we can adapt pretty fast to the new requirements.

We can deliver usable software, quickly.

Frequent customer feedback.

2 Case 2

2.1 Ideas

- Scrum
- Lean Design

2.2 Decision

The main concept is, we are two (CS?) students with an idea for a mobile app, which should become really successful.

We decided against *Scrum*, because there we need a large development team, which actually does not exist, because we have a really small team.

Therefore, we concluded that the best solution for our project is to use an agile software development process called *Lean Development Process*, because we are focused on customer interaction and the whole process is inexpensive and is good for small development teams.

2.3 Final words

For our case the Lean Development Process fits in best way to our scenario. But we came to the point, that every agile development process would fit to our case.

3 Case 3

3.1 a)

According to the needs of the customer and the lively competition, we are obliged to be very efficient. Another factor is that the customer has a list of requirements in form of a list, that will most likely have to be changed in the future.

We decided to use an agile software process. The customer will most likely make changes to their original list of requirements during development, so we have to be able to adopt to those changes. An agile software process will allow us to be flexible enough to react to changes to the requirements.

A problem is that we are not able to make a perfect work schedule. The object-oriented design will allow us to change the already existing software more efficient.

3.2 b)

First we will ask the customer for the list of the additional software functionality, so we can get an idea of the scope of the project.

Then we will approach the customer to talk with them about their requirement list. After that, we will get the necessary knowledge about their software. The next step is to make a price offer and make a contract with the customer. Then we will give them feedback and will continue with feedback through the work process.

We will create short sprints, in which we will implement a few features and will show the customer the software after each increment, get additional feedback and update the requirements list according to that feedback.

4 Case 4

We think the *incremental software process model* would be the best fit for this situation, since we are very pressed for time with our hard deadline before the holiday season.

4.1 Advantages

Using the incremental software process model, we don't need to do extensive analysis of our requirements at the beginning, so we save time.

Also we don't need to do a lot of documentation, which also saves us time.

Another benefit is the ability to test each incremental version for frequent customer feedback, which helps to see if the game is fun to play and validate added features.

If we don't get all the features implemented before our deadline, we would also still be able to deliver the game on time, since each increment of the game is already playable. We could still add more features through updates, later on.

4.2 Disadvantages

The main disadvantages of this process do not affect us that much. First of all, we already successfully developed the prequel game, so we are already experienced and have a good understanding of the system architecture we need, to support our necessary functionalities.

Our games also doesn't need to meet any external regulations or certifications, so that's not a problem either.