

SSW 555: Agile Methods for Software Development

Homework 1: SDLC methods for Self-Driving Cars

1)Waterfall Model:

Waterfall model divide in different separate parts which are basically five phases in this software development process. In this process the result of the previous step is taken as a initial stage of the next phase. Basically, this model use for large group of projects.

Process:

-In the first phase, it will gather all the important requirement of self-driving car such as amenities of the car which will help to support autonomous cars. Moreover, amenities of driverless cars may include navigation, speed limit, sensors which is for avoiding accidents, parking and other important requirement. In the second step, the system is prepare the design of the model which help us to define which part goes to hardware and which is for software side. After that, it implements all the parts of the system which define in the previous stage. Furthermore, forth phase is testing stage where we can verify the functionalities whether it is working properly as we decided earlier. Finally, in the final stage which is maintenance. In that it requires up-to-date maintenance and examine the car whether it performing as they required to.

Advantages:

- This model is rigid model. individual phase has equal importance that is the reason the chance of failure is low.
- Every stage it checked that requirement are fulfil as expected before moving to another phase.
- Each and every stage define clearly and deeply which helps to understand easily.

Disadvantages:

-It is very difficult change the requirement after the first step because we have to go back to first step and define each and everything again. So if new amenities come in driverless car it create problem with previous requirement.

Priority: - I would like give this model second priority because it limits the phase to re-revise the thoughts. Under uncertain cases the project has nothing to deliver.

2)Rational Unified Process (RUP)

RUP model divided in to four steps. Each and every steps involve business modeling, analysis and design, implementation, testing, and deployment. After each and every step progress goes to client and they can review and make a decision whether it can go to next step or not.

Process:

-In the first phase which is inception it decide the budget and project is worth it or not. The system is decide what resources is used for self-drive cars. And make basic use case diagram. Moreover, in the second step, detail requirement of resources and creator consider estimate application of the software which help to decrement the risk which called elaboration. After that, the project is implemented and tested in the construction phase. In the last phase model is go to public and base on feedback from client it can update.

Advantages:

-RUP allow us to make any changes in the process in any phase whether it comes from client or project itself.

-This model may require more time but during the project bugs is less. Also, at the end of the project there is less error.

Disadvantages:

-It is complex process for small project because its implementation is challenging.

Priority: - Highest priority would be given to this model. Because, the risk factor is reduce as we can re-iterate any phase at any point of time in developing automated car software. At the end though it takes more time we can deliver the software with its best feature.

3)Extreme Programming (XP)

Extreme programming has main focus is to give good quality of the software. They more focus on coding. There are 12 phases.

Process:

1. The Planning Game: - business people decide scope priority and release date. However, technical team decide technical cause, process and detail scheduling.
2. Small Releases: -each and every release should contain valuable business features and it should be small as possible as they can. Each release must have its new feature of implementation.
3. Metaphor: - It is simple explanation which is accept by all the member of the group. Easily understandable by customer.
4. Smile Design: - It has to run successfully in any tests and has no duplicate logic require.
5. Testing: - Programmer and customer need confidence in right operation.
6. Refactoring: -rewrite the code for improve better version of the code and always find out simple version of program. Always find out low risk of way.
7. Pair Programming: - think about best way to implementation which two people can write the code in the same machine at a time. Frequently change the role of the programmer.
8. Collective Ownership: -any of the member of the project they can improve the productivity at any time.
9. Continuous Integration: - every one-hour programmer has to test the program and if nay code broke it has informed.
10. Sustainable: - each person has to fresh every morning. Overtime allow when serious problem occurred.
11. Whole Team: - customer is part of the group so programmer can ask any question to the customer.
12. Coding Standards: - standard should go along with group. Everyone can edit anyone's code during making model.

Advantages:

-XP model save more time and cost because this process only focus on time to time delivery.

-Main advantage is that developer can improve the code in any time.

-XP model do regular testing that the reason the final product is faster

Disadvantages:

-This model only focus on coding rather than design.

-XP model only applicable for small project and low budget. It is not good for large scale projects

Priority: -I would like to give last priority to this model as there is so much liberty of restructuring the code creates unusual and complex situation.