Homework 9

Roles in FDD (Feature Driven development)

1) **Project Manager**: It is a duty of Project Manager to share the progress report to the client and make sure that project will run on right and proper direction. They can handle multiple projects at a time.

2**) Chief Architect**: Chief Architect is the one who responsible for the design of the system and whole team of designer is handled by him/her. For small scale project, it can be one person as well.

3) **Development Manager**: Development Manager can handle more than one team or projects at a time. Development manager handles the whole development team, and it is his/her responsibility to make sure that all the team member can finish their task on time.

4) **Chief Programmer**: The most experienced person in all the programmers is the Chief programmer. If developer need any help in programming, then it is a duty of chief programmer to help them and make sure that project will run on right path or direction. Only one particular project handled by the chief programmer at a time.

5) **Class Owners**: The smallest bunch of feature development is defined as the class which develops in maximum fourteen days or two weeks. They are the developer who develop the all the features. They can take the help for the problem or guidance from the chief programmer and send the report which shows the progress of the project to the development manager.

6) **Domain Experts**: The Domain experts is one of the persons who has the deep and highest knowledge about the domain, and he/she can be able to helps the team to get the understanding of it.

Give an example of each FDD role's responsibilities in the new horn honking feature.

• **Project Manager:** He/she determines the budget for each feature and required equipment such as for horn honking feature like sensor, speakers, etc.

• **Chief Architect**: Plans how is the horn honking feature is looks like.

• **Development Manager:** For the horn honking feature, He/she can make sure that all the team member can complete a task on time.

• **Chief Programmer**: He/she is responsible for the management of the whole development team and analyze of the feature horn honking and make sure that it will be implemented as right way.

• **Class Owners**: They are responsible for the successful horn honking feature implementation.

• **Domain Experts:** They are responsible to understand the horn honking feature deeply to the client or customer.

Corresponding role to extreme programming.

Extreme Programming is an agile software methodology which helps gaining the best quality of the software, which is designed by the development team. When the development team is small the work assigned is bound to change and distributed according. The team works collaboratively and communicate face to face. They aim at finish the task assigned. There is no quality change. The features, cost and requirements are taken almost care of. In Extreme programming the priorities are decided by the end user. It takes 1 to 2 weeks to complete the user story.

FDD is suitable for long term projects like the driverless cars. They work in delivering the customer satisfactory product. They make an appropriate workflow to accomplish the task. In FDD the small deliverables are given to the customer within 2 to 10 days. More documentation as compares to XP.

Advantages of FDD:

1) FDD helps in achieving the task in easier way within short period.

2) FDD is flexible in attaining its goal and makes sure that the product delivery is fast.

3) FDD system has fewer base stations. It manages progress tracking and reporting capabilities for the driverless cars.

4) There are five methods which helps in accomplishing the task.

5) FDD is scalable and is can handle the complex work for the driverless cars.

Disadvantages of FDD

1) The channel reciprocity is easy in TDD as compares to FDD.

2) FDD has less documentation so there is no proof for the idea and becomes difficult to make changes in the design if required in future for the car.

3) FDD has not much written documentation, so the customer has not much idea about the driverless cars.

4) FDD cannot be used for smaller projects and small organizations.

5) In FDD the frequencies are allocated in the fixed manner which leads to the wastage in the spectrum.

6) Iterations is not well-structured like in other agile frameworks.

7) The work is done in an isolated manner in compared to XP.

8) All the tasks performed are under the supervision of the developers so need to coordinate with them.

FDD is very useful for the large projects like the driverless cars. It becomes very convenient for such projects. The workflow becomes easy going. It has multiple teams to build the various features for the cars. The project divided in small chunks help in faster delivery of the car. The end users can ask for the improvement wherever they expect for. There is full involvement of the customer. Customer satisfactory product is deployed and delivered within the time boundaries thus FDD is recommended.