The projects consist of people who are the most important aspect. The process is defined by people and the technology is used by people. The Feature Driven Development(FDD) defines six key roles and implies some additional roles.

**Key Roles of FDD:**

The Project Manager (PM) is the administrative head of the project. PM is the sole responsible for reporting progress in the project, manages equipment, space and resources, manage budget and more.

The Chief Architect (CA) is the designer of the overall system. He runs the design session or workshop where the team collaborates in the design of the system. The CA requires technical, modelling and facilitation skills. They also steer the project through the technical obstacles.

The Development Manager (DM) is responsible for leading the development activities day-to-day. They require facilitation and good technical skills. He/she resolves the technical conflicts.

The Chief Programmers are experienced developers. They are included in high level requirement analysis and design activities of the project. They lead small teams of 3 to 6 developers.

The Class Owners are developers who are sole responsible for a class. They design, code, test and document the features they create and maintain for the new software.

The Domain Experts are users, business analysts, stake holders or mix of these. The developers rely on them for the detailed requirements. They need good verbal written and presentation skills. They are very critical to the success of the system.

As per the FDD, **process 1, first the overall model is developed**. In this, the domain experts and chief programmers are formed into a modelling team. The team is led by Chief Architect. They perform domain walk-through by discussing about the requirements. The basic requirement will be the horn honking during certain situations like **when the sensor senses a vehicle or pedestrians within a specific distance that is very close, it must honk once or twice but not continuously. Another requirement would be to honk in sharp turns where it cannot see at least 200 feet ahead.** These requirements are provided in high level by the users and it is further analysed by the business analyst for feasibility. They study relevant documents to understand completely. A model is developed and refined. The notes are written about the model and assessed by internal and external assessment team.

In **process 2**, **features list is built.** The chief programmers form together to define the overall features. They are called features list team. The list is hierarchical and small enough to implement in 10 days. A template is used to describe each feature. **<Action><Result><Object>. Example, Honk the Horn during Sharp turns.**

In **process 3, planning by feature**. The planning team is formed by the development manager and chief programmers. They determine the sequence of development and assign **business activities to Chief programmers and classes to developers**.

In **process 4, design by feature,** the chief programmers and developers form a dynamic team for each feature. They are known as Feature team. In this phase, all the people in the team have roles and responsibilities. **Domain walk through by domain expert, Study documents by team, Develop UML diagrams by the team, Refine the object model by chief programmer, Classes and methods design by developers and the design inspection by the team.**

In **process 5, build by feature,** the class owner implements the classes and methods. The peer review or code inspection is done by the feature team. Unit testing is done by the class owner. Once the tests are successful, the class owner with chief programmer promote to the build. Finally, they disband the teams.

When the process is compared to XP, XP will have more details while FDD will have an overall prototype. FDD build features in hierarchical steps whereas XP will have a flat structure. FDD planning team build by feature and XP build by sprints. Unlike XP’s collective ownership, classes are owned by individual developers. A dynamic team is formed to build feature in FDD. In contrast, XP has a team formed already at the beginning. FDD uses peer reviews and code inspection over Pair programming. FDD’s main aim is Up-front modelling – “JEDI(Just Enough Design Initially)”. It avoids refactoring. In contrast, the plan driven approaches have BDUF(Big Design Up Front).

I would recommend Feature Driven Development rather than XP. In my point of view, the product development project must use FDD and service implementation can use XP approach. FDD provides a visual experience of product to the users through which they will be almost satisfied. They can decide right away. The feedbacks are received immediately. Since, the XP approach have more of documentation and no prototype, it is suitable only for services.