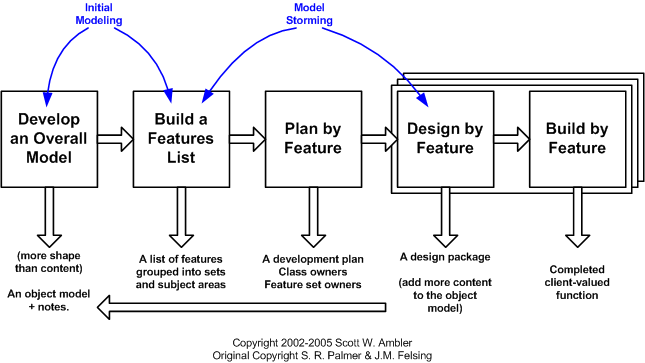
**Name:**Vaishnavi Shankar Bhambure

**Software Testing\_7670**

**Feature Driven Development (FDD)**

This method is focused around “designing & building” features. Unlike other Agile methods in software engineering, FDD describes very specific and short phases of work that has to be accomplished separately per feature. It includes domain walkthrough, design inspection, promote to build, code inspection and design. FDD develops product keeping following things in the target

1. Domain object Modeling
2. Development by feature
3. Component/ Class Ownership
4. Feature Teams
5. Inspections
6. Configuration Management
7. Regular Builds
8. Visibility of progress and results



**FDD has five key activities that we can't find in Scrum.**

* **Develop the overall model**– the team will determine the project scope. They will propose multiple models and create one to top them all.
* **Set up a feature list** – outline the customer-focused features that need to be developed. For example, small functions that can be finished in a short time.
* **Plan by feature** – the team assesses individual features and arranges them properly. Then, these features will be assigned to different team members.
* **Design by feature** – the chief programmer will select which features to develop in the next two weeks.
* **Build by feature** – the final stage. Developers work to develop a code for previously mentioned features. This code is tested before creating the final version.

**ADVANTAGES OF FEATURE DRIVEN DEVELOPMENT:**

•Feature-Driven Development helps to move larger size projects and obtain repeatablesuccess.

•The simple five processes help to bring work done in a short time and easiest manner.

•This type of model is built on set standards for software development industry, so it helpseasy development and industry recognized best practices.

•Feature driven development offers well defined progress tracking and reportingcapabilities

.•FDD also support multiple teams working in parallel.

**Disadvantages of FDD Methodology:**

•Not an ideal methodology for smaller projects so, it is not good for an individual softwaredeveloper.

•High dependency on the main developer means the person should be fully equipped foran act as coordinator, lead designer, and mentor.

•No written documentation provided to clients in this methodology so, they are not able toget a proof for their own software.

•Promotes individual code ownership as opposed to shared team ownership.

•Iteration are not well defined by the process as other agile methodologies.

•The model-centric aspect can have hug impacts when working with an old system thatdoes not have a system model