Week09 – Short Paper Assignment – Feature-Driven Development

SWEN 603 9041

Debashis Jena

University of Maryland Global Campus

# Introduction

Before we go deep into feature-driven development, first thing we need to understand what is a feature. Basically, a feature can be any part of a system that can be independently developed and accessed. For example, in facebook.com website, the login to the application itself is a feature, similarly add friend is another feature. In a traditional project, the work breakdown structure is created to split a system into multiple sub-systems. Thus, it makes the project estimation easier. Similarly the feature driven development is a method that breaks the whole application into smaller pieces and then each of the piece is further split into features. Since this is a agile method, it follows agile manifesto and the principles. FDD is also an incremental and iterative development process. Each iteration may consist of the development of one or multiple features. Teams try to develop multiple features at the same time and integrate at the end of the iteration.

Feature driven development was initially created by Jeff De luca in 1997 at a bank. This project was to create a system for the bank and had a time line of 15 months and staffed 50 members. In this project he followed the process of modeling, listing, planning, design and building of the features. Later in 1999, Jeff De Luca mentioned the details about the process in the book “Java Modeling in Color with UML” and that’s where the process officially came into picture.

# Overview

As agile modeling website states, “Feature-Driven Development (FDD) is a client-centric, architecture-centric, and pragmatic software process.”. As the name suggests, features are the most important topic of FDD. A feature is a small, customer-valued function that can be expressed in the form of action, result and object. A few examples of features in a streaming application can be as follows.

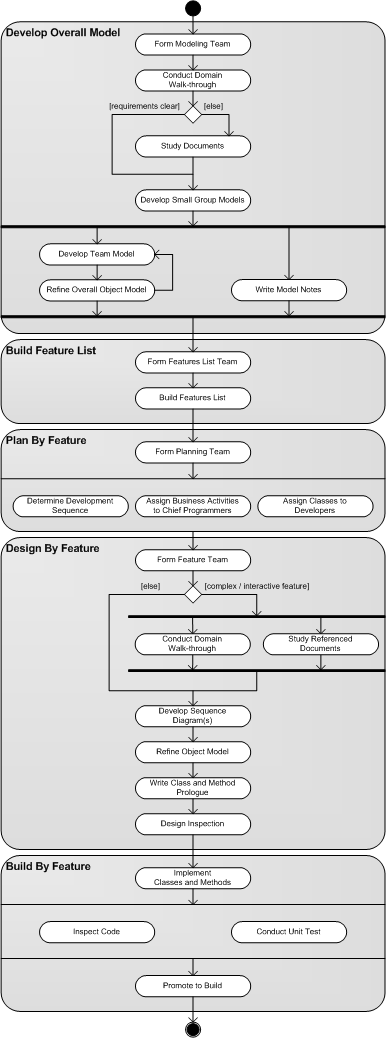
1. User authentication using username and password.
2. Get the customer subscribed video plan from the database.
3. Display the profile to be selected.
4. Display featured videos.
5. Categorize videos by different genres and languages.
6. Map videos with their subtitles.

The feature driven development goes through a set of phases or activities, which will be described in the following section.

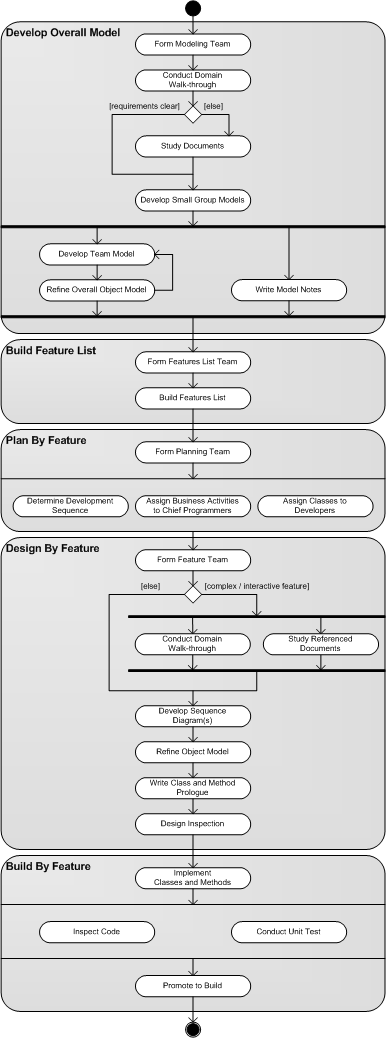
**FDD Project Lifecycle**

The goal of each and every project is to successfully deliver the project by meeting customer’s requirements and acceptance criteria. The success criteria of the project may be based on completion of each of the features within the system. Only when each of the features are functional, then the implementation of the full application can be considered to be complete and the project may be considered as successful. The feature-driven development process tries to ensure each and every features within the system are addressed and implemented as per the requirement. Thus, the FDD project lifecycle process goes through five primary activities.

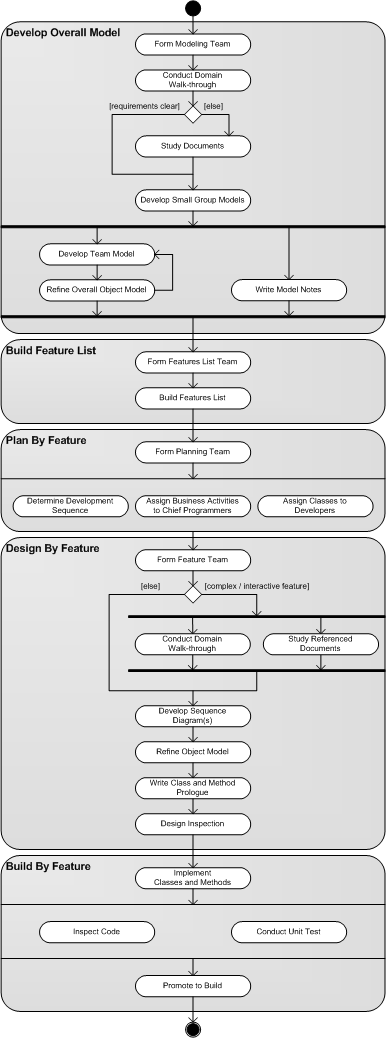
1. ***Develop overall model***

This is the definition phase of the project. This activity consists of sub-activities; like form modeling team, conduct domain, walk through, study documents, develop small group models, team model, write model notes. The diagram displays the work flow from one sub activity to the other. The goal of this phase is to come up with the overall model of the final product.

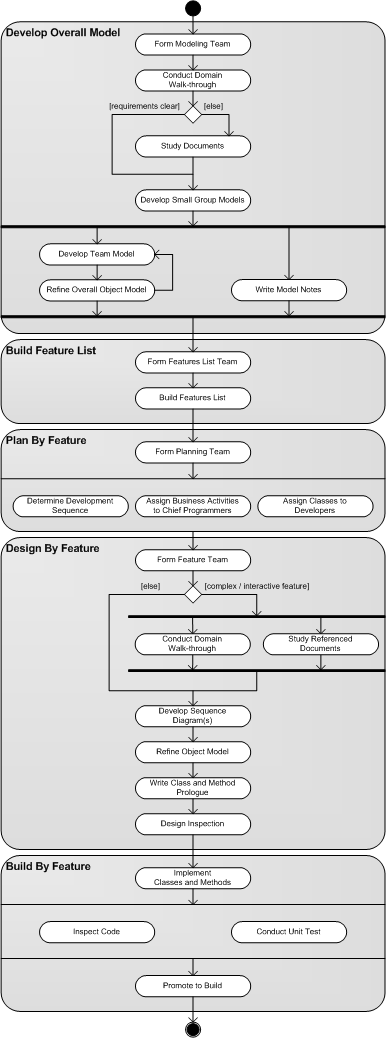
1. ***Build feature list***

As the overall model is complete, now the task is to extract the features that can be individually built in a short time and with the help of a small team consisting of not more than 3 members. First task in this phase is to form the features list team, which is comprised of the chief programmers and modelling team. This team will decompose the system to small and simple functional subject areas.

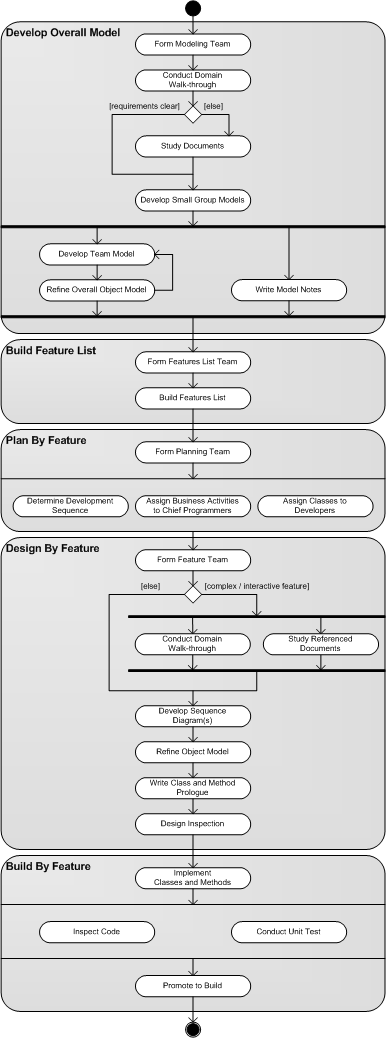
1. ***Plan by feature***

The expected output of this phase is to come up with the plan how to go with the list features, which should be developed first and the prioritization of the features. Typically the planning team is comprised of the development manager and the lead developer(s). As mentioned earlier, the sequence of development needs to be determined in this phase. The lead programmers are given the ownership of these features and the business activities.

1. ***Design by feature***

This phase is repeated for each of the features. The full design needs to be created for each of the features. A design is similar to creating an architecture but in the smaller size. It is called a design package. As the chief programmer is assigned in the previous phases, the features are identified that needs to be implemented in two week period. The feature may be further decomposed to different classes and may be assigned to the team members in the feature team. The assignees are called class owners. The contribution of the class owners is the same as the chief programmer in this phase. The class and method definitions are documented and then a design inspection may be incorporated in this phase. There may be a peer review between the class owners and the chief programmer.

1. ***Build by feature***

This is the actual implementation phase of the project. Each of the features are built and then merged to the master build. Also, unit tests are one of the integral part of the implementation phase. They make the class owners gain confidence on the classes and methods they build. As the features are built completely, a feature specific testing may take place. So, this phase goes through the below sub activities.

* Implement classes and methods
* Inspect code
* Unit tests
* Promote to build

**Advantages and Disadvantages**

As FDD works best when the features are listed taking every considerations of the system requirements. The process also has some drawbacks. Here are a few of the advantages and disadvantages of the FDD method.

***Advantages***

1. Gives the project team a great understanding of the project and it’s scope.
2. User-centric approach
3. Suitable for large scale and long term projects.
4. Smaller features gives the project manager an easy control.

***Disadvantages***

1. Not ideal for small scale projects.
2. Highly dependant on the chief programmers who drive the features.
3. Lacks documentation as compared to the other methods.
4. Class ownership may lead to possible isolation of the code knowledge.

**Reference**

Lucidchart Content Team. (n.d.). *Why (and how) you should use feature-driven development.* Retrieved from *https://www.lucidchart.com/blog/why-use-feature-driven-development*.

Saleh, Hala. (2015). *Agile project management fundamentals: fdd processes and tools*. [Books24x7 version] Available from *http://library.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=87670*.

De Luca, J.. (2003). *De Luca on fdd – newsletters*. Retrieved from *http://www.featuredrivendevelopment.com/forum/2*.