FEATURE DRIVEN DEVELOPMENT METHODOLOGY

Privacy - Terms

February 28, 2018 | Management |

In our previous articles, we talked about software development methodologies that help optimize workflows. Today we will talk about Feature Driven Development, evaluate the pros and cons of this approach and take a look at the experience of organizations that use them to help you optimize processes.

Feature Driven Development methodology (abbreviated FDD) was developed by Jeff De Luca and a recognized guru in the field of object-oriented technologies Peter Coad. Like other adaptive methodologies, it focuses on short iterations, each of which serves to work out a certain part of the system's functionality. According to FDD, one iteration

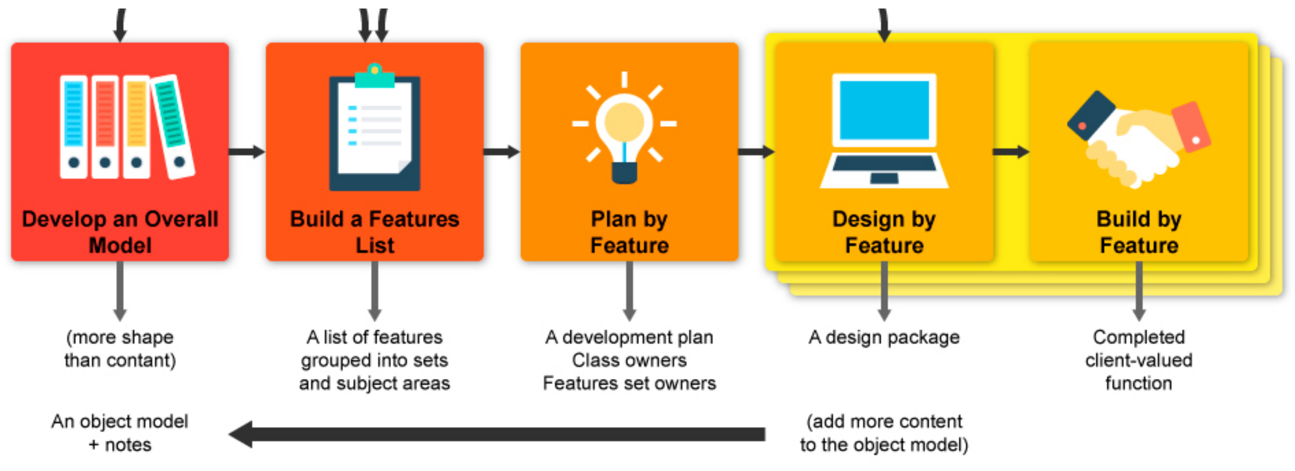
Briefly about FDD

In general, the approach can be described by the following sequence:

1. Develop an Overall Model
2. Build a Features List
3. Plan by Feature
4. For each property – Build and Design by Feature.

Ideally, in the end, we get a platform, gradually overgrown with properties.

Privacy - Terms

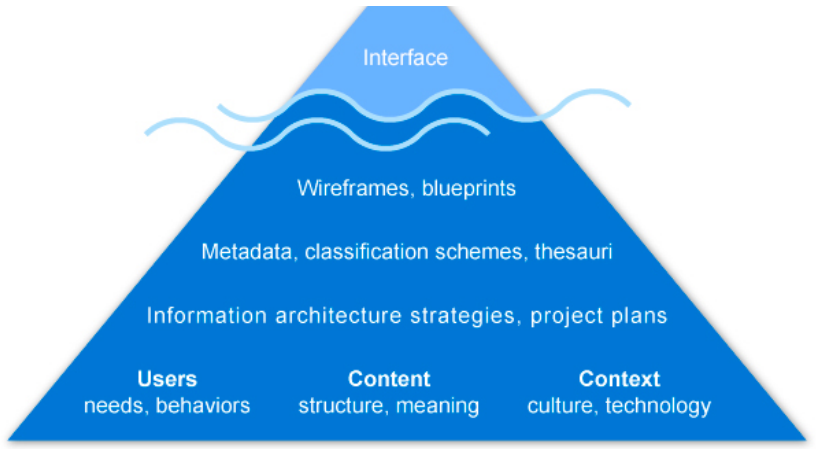

Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than contant)
A list of features grouped into sets and subject areas
A development plan Class owners Features set owners
A design package
Completed client-valued function
An object model + notes
(add more content to the object model)

Develop an Overall Model

According to Jeff De Luca, the general model is more a draft or outline than a detailed content. However, it is not enough just requirements for the project to get this model. The model is a vision of the project, based on the results of preliminary research. Thanks to Louis Rosenfeld, we know how much information architecture is important in web development. At the stage of the general model, we should have an analysis of the target audience, analysis of the content and context of its use, the content structure, including metadata and thesaurus. There must be a document describing the principles of the user interface, a list of used UX components, a description of their properties and behavior.

In this approach, it is understood that we are starting to develop, with only a rough outline of the system, which, as it develops, grows in details. Thus, the detailed

Privacy - Terms



wireframes will be developed to create the appropriate properties or, in other words, parts of the project. However, at the stage of the general model, the “raw” skeletons of the main interfaces and the preliminary site map will still be required.

All the above described are related documents. The main result of the stage is the domain model. This is a static UML diagram that includes all the project’s logical objects and their interaction (the object is part of another object, the object extends another object, etc.).

Build a Features List

We do not think that here it is worth going into details. Most of us use SCRUM methodology and a feature list in the FDD – the same as the product backlog in

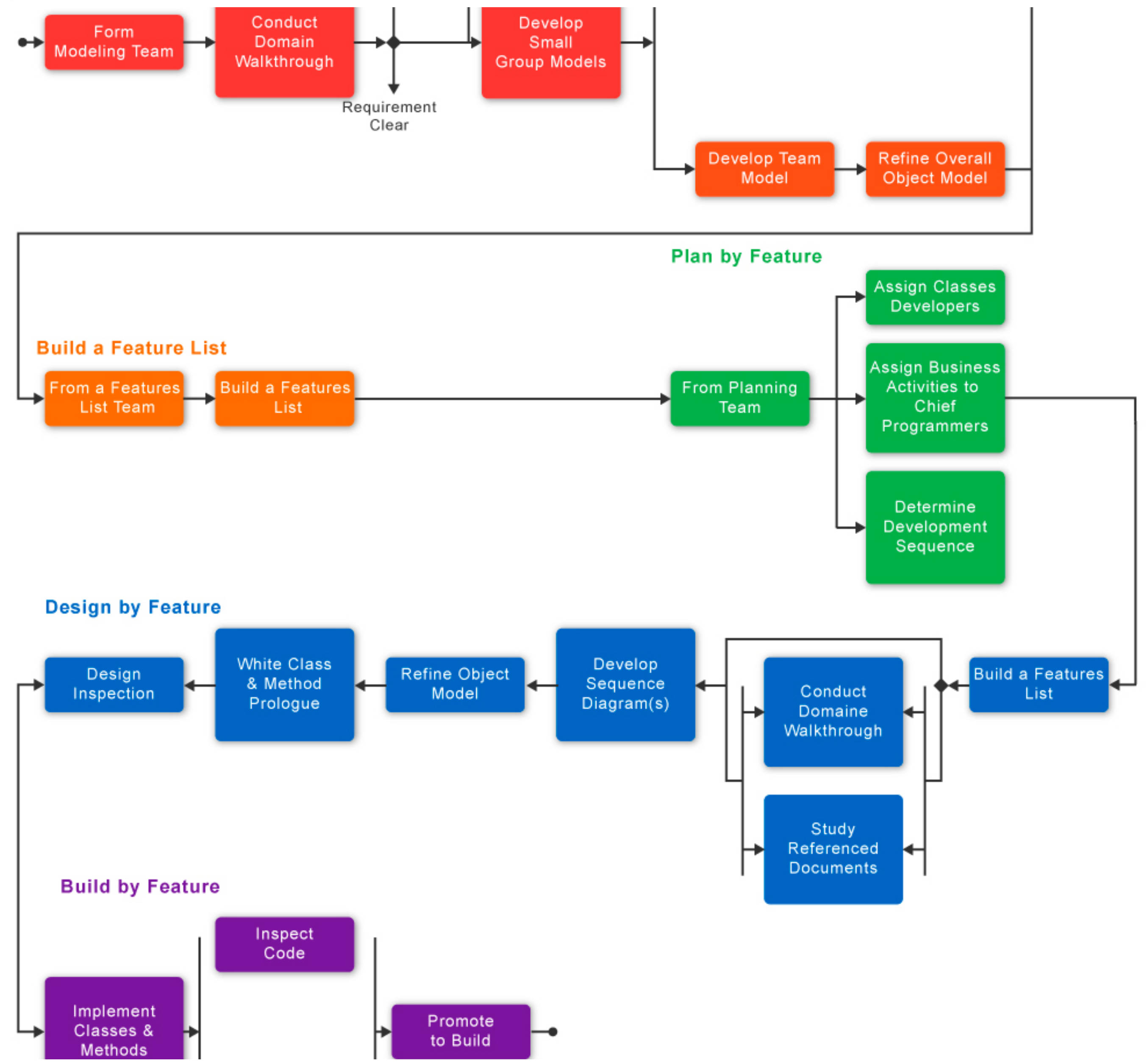
Plan by Feature

It seems to be easy and clear, but how to estimate the time of development of the properties of the project, when the details are not specified. Related tasks for developers will be very clear only when all the preliminary work on this property is completed. How to plan? And then Story Point Estimation comes to the rescue. We estimate properties not in man-days, but in abstract points as the property is more complex. From the entire list it is proposed to select a property of minimal complexity and assign it 2 points of difficulty. It is recommended to use an exponential scale, for example: 1, 2, 3, 5, 8. The question then arises, why we have not assigned 1 to trifling property. Just in case. The list of properties is huge and at first glance it is unlikely that you will be as accurate as possible in the assessment.

All members of the team take part in the evaluation. That is, each property is discussed from the perspective of all development stages until the team comes to a single assessment of complexity. Tedious, but productive.

Design by Feature

So, the priorities are defined, the feature is selected for implementation. It's time to describe the requirements. To do this, a Use Cases model is created with respect to the selected property. This document would be good to provide an Activity Diagram for visualizing the logic of conjugate usage scenarios. The use cases themselves can be reflected in the appropriate UML diagram (Use Case diagram). All members of the team take part in the review of the document.


Form Modeling Team
Conduct Domain Walkthrough
Requirement Clear
Develop Small Group Models
Develop Team Model
Refine Overall Object Model
Plan by Feature
Assign Classes Developers
Build a Feature List
From a Features List Team
Build a Features List
From Planning Team
Assign Business Activities to Chief Programmers
Determine Development Sequence
Design by Feature
Design Inspection
White Class & Method Prologue
Refine Object Model
Develop Sequence Diagram(s)
Conduct Domaine Walkthrough
Build a Features List
Study Referenced Documents
Build by Feature
Inspect Code
Implement Classes & Methods
Promote to Build

Privacy - Terms

Further work can be carried out in parallel: part of the group is engaged in technical design, some work on frameworks (or on the detailing of frameworks if they have already been prepared at the stage of the general model).

The technical design involves a UML diagram with the domain model entities, a list and a class diagram (where relevant), coupled with the model property, DAO, controllers, services, decorators (possibly helpers). There is also a DB architecture covering the property in the form of an “entity-relationship” model (ER Diagram). If the property affects any external systems (for example, the user preferences profiler), this should also be reflected in the technical design, for example, in the form of a Sequence diagram.

Among other things, you should prepare Test Cases, which will allow QA-experts to fully appreciate the quality of the implementation of the property.

When the domain model entity diagram for the selected property is finished, details that are not considered in the general model are displayed. So it should be: you simply reflect new details on the general model as they appear.

At the end of the stage, the entire development team takes part in the technical design review. In addition to “polishing” all aspects of the document, it provides a deep insight into the ideas of the architecture of all project developers.

reviewing the visual design and prototype properties, xHTML-imposition and modification of decorators (views) are performed. Surely the creative genius of the visual designer will also require additional programming on the client side (JS). When ready property, it is transferred to the QA-department for testing. After the property is tested and gone into the product, we take the next priority property and repeat the design / implementation cycle.

Bottom Line

The success of the project depends on the experience and knowledge of the lead programmer. In FDD, he plays all the main roles: leader, mentor, analyst and so on. However, the methodology was created for long-term projects, therefore, as residents of Stack Overflow note, it is useless to apply it for small tasks.

However, there are advantages. Continuous reporting on progress at all levels helps to track progress and results. This allows you to regularly update the project, identify errors and provide the client information at any time. And one of the residents of Stack Overflow claims that the main advantage of FDD is the ability to assess at any time whether the project lags behind the schedule or moves faster.

As it was already mentioned, FDD is used in large-scale projects, since at the first stage a common model is being developed, which makes it possible to understand the product. This same property helps to attract new developers. At the same time, a

Privacy - Terms

2018 >>

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Contact Us

Our Offices

Privacy - Terms

Email us:

Ukraine

Bulgaria

United States

New Line Technologies

Turkey

1288. Sk. Sinan Mahallesi

Muratpaşa, ANT 07100

+90 537 849 0577