FDD Process #1: Develop an Overall Model

A initial project-wide activity with domain and development members under the guidance of an experienced object modeller in the role of Chief Architect.

A high-level walkthrough of the scope of the system and its context is performed. Detailed domain walk-throughs are then held for each area to be modelled. After each domain walkthrough, small teams are formed with a mix of domain and development staff who then compose their own models in support of that domain walk-through. The teams each present their models for peer review and discussion. One of the proposed models, or a merge of the models, is selected by consensus thus becoming the model for that domain area. A merge of the domain area model into an overall model is performed, adjusting model shape as required.

The object model is then iteratively updated with content by the Design by Feature process #4.

Entry Criteria

- Domain experts, Chief Programmers and the Chief Architect have been selected.

Tasks

Form the Modelling Team	Project Manager	Required

The modelling team comprises permanent members from the domain and development areas, specifically the domain experts and the chief programmers. Other project staff are then rotated through the modelling sessions so that everyone gets a chance to participate and to see the process in action.

Domain Walk-through	Modelling Team	Required

A domain expert gives an overview of the domain area to be modelled. This should also include information that is related to this domain area but not necessarily a part of its implementation.

Study Documents	Modelling Team	Optional

The team studies available reference or requirements documents such as object models, functional requirements (traditional or use-case format), data models, and user guides.

Develop the Model	Modelling Team in Small Groups	Required

Forming groups of no more than three, each small group will compose a model in support of the domain area. The Chief Architect may propose a "strawman" model to facilitate the progress of the teams. A member from each small group presents that groups proposed model for the domain area. The Chief Architect may also propose further model alternatives. The modelling team selects a proposed model or composes a model by merging ideas from the proposed models.

Refine the Overall Object Model	Chief Architect, Modelling Team	Required

Every so often, the overall object model is updated with the new model shapes produced by iterations of the Develop the Model task above.

Write Model Notes	Chief Architect, Chief Programmers	Required

Notes on detailed or complex model shapes and on significant model alternatives are made for future reference by the project.

Verification

Internal and External Assessment	Modelling Team, Business	Required

Internal or self-assessment is achieved by the active participation of domain experts. On an as needed basis, external assessment is made by referring back to the business (users) for ratification or clarification of issues that affect the model.

Exit Criteria

The result of the process is the object model

- Class diagrams focusing on model shape. That is, what classes are in the domain, how are they connected to one another and under what constraints.
- Methods and attributes identified are placed in the classes.
- Sequence Diagram(s), if any.
- Model notes to capture why a particular model shape was chosen and/or what alternatives were considered.

FDD Process #2: Build a Features List

An initial project-wide activity to identify all the features to support the requirements.

A team usually comprising just the Chief Programmers from process #1 is formed to functionally decompose the domain into Subject Areas, the Business Activities within them and the Steps within each Business Activity, thus forming the categorised features list. The top-level categorisation for the features list comes from the partitioning of the domain by the domain experts in process #1.

Entry Criteria

- Domain experts, Chief Programmers and the Chief Architect have been selected.

Tasks

Form the Features List Team	Project Manager, Development Manager	Required

The team comprises the chief programmers from the modelling team in process #1.

Build Features List	Features List Team	Required

The team shall identify the set of features using the knowledge obtained from process #1. This is simple functional decomposition into subject areas that comes from the partitioning of the domain by the domain experts for their domain area walkthroughs in process #1. It is decomposed into subject areas that comprise business activities that comprise business activity steps (features). Features are granular functions expressed in client-valued terms using this naming template:

> <action> <result> <object>
>
> For example, calculate the total of a sale, calculate the total quantity sold by a retail outlet for an item description

Features are granular in accordance with the rule that a feature will take no more than two weeks to complete, but not so granular as to be at the level of getters and setters. Two weeks is an upper limit; most features take less than this time. When a business activity step looks larger than two weeks, the step is broken into smaller steps that then become features.

Verification

Internal and External Assessment	Features List Team, Business	Required

Internal or self-assessment is achieved by the active participation of modelling team members. On an as needed

basis, assessment is made by referring back to the domain experts from the modelling team or the business (users) for ratification or clarification of issues that affect the features list.

Exit Criteria

The result of the process is the Features List

- A list of subject areas
- For each subject area, a list of the business activities within that subject area
- For each business activity step, the feature to satisfy the step

FDD Process #3: Plan By Feature

An initial project-wide activity to produce the development plan.

The project manager, development manager and the chief programmers plan the order that the features are to be implemented, based on feature dependencies, load across the development team and also on the complexity of the features to be implemented. The main tasks in this process are not a strict sequence. Like many planning activities they are considered together, with refinements made from one or more tasks and then considering the others again. A typical scenario is to consider the development sequence, then consider the assignment of business activities to chief programmers and in doing so, consider which of the key classes (only) are assigned to which developers (remember a chief programmer is also a developer). When this balance is achieved and the development sequence and assignment of business activities to chief programmers is essentially completed, then the class ownership is completed (beyond the key classes that were already considered for ownership).

Entry Criteria

- The Build a Features List process has completed.

Tasks

Form the Planning Team	Project Manager	Required

The planning team comprises the development manager plus the chief programmers.

Determine the Development Sequence	Planning Team	Required

The planning team shall assign a date (month and year only) for completion of each business activity. The identification of the business activity and the completion date (and thus the development sequence) is based on

- Dependencies between features in terms of classes involved
- Balancing load across class owners
- The complexity of the features to be implemented
- Bringing forward high-risk or complex business activities
- Consideration of any external (visible) milestones such as betas, previews, feedback checkpoints and the "whole products" that satisfy such milestones

Assign Business Activities to Chief Programmers	Planning Team	Required

The planning team shall assign chief programmers as owners of business activities. The assignment is based on

- The development sequence
- Dependencies between features in terms of classes involved
- Balancing load across class owners (remember that chief programmers are also class owners)

- The complexity of the features to be implemented.

Assign Classes to Developers	Planning Team	Required

The planning team shall assign developers as class owners. Developers own multiple classes. The assignment of classes to developers is based on

- Balancing load across developers
- The complexity of the classes
- The usage (e.g. high-use) of the classes
- The development sequence

Verification

Self Assessment	Planning Team	Required

As the planning is a team activity, a self-assessment is achieved by the active participation of chief programmers, development manager and the project manager.

Exit Criteria

The result of the process is the development plan consisting of

- Business activities with completion dates (month and year)
- Chief programmers assigned to business activities
- Subject areas with completion dates (month and year) derived from the last completion date of their respective business activities
- The list of classes and the developers that own them (the class owner list)

FDD Process #4: Design By Feature

A per-feature activity to produce the feature design package.

A number of features are scheduled for development by assigning them to a Chief Programmer. The Chief Programmer selects features for development from his "inbox" of assigned features. He may choose multiple features that happen to use the same classes (hence developers). Operationally, it is often the case that "sets" of features are scheduled for development at a time by the Chief Programmer. Such a set is called a Chief Programmer Work Package.

The Chief Programmer then forms a Feature Team by identifying the owners of the classes (developers) likely to be involved in the development of the feature(s) he selects for development. This team then produces the Sequence Diagram(s) for the assigned feature(s). The Chief Programmer then refines the Object Model based on the content of the sequence diagram(s). The developers then write class and method prologues. A Design Inspection is held.

Entry Criteria

- The Planning process has completed.

Tasks

Form Feature Team	Chief Programmer	Required

The Chief Programmer identifies the classes likely to be involved in the design of this set of features and updates the feature database accordingly. From the class ownership list, the Chief Programmer identifies the developers needed to form the Feature team. As part of this step, the Chief Programmer creates a new Design Package for the feature(s) as part of the Work Package.

Domain Walk-through	Domain Expert	Optional

The domain expert gives an overview of the domain area for the feature to be designed. This should also include domain information that is related to the feature but not necessarily a part of its implementation. This is an optional task based on the complexity of the feature and/or its interactions.

Study the Referenced Documents	Feature Team	Optional

The feature team studies the referenced document(s) for the feature to be designed, all confirmation memos, screen designs, external system interface specifications and any other supporting documentation. This is an optional task based on the complexity of the feature and/or its interactions.

Develop the Sequence Diagram(s)	Planning Team	Required

Develop the Sequence Diagram(s) required for the feature to be designed. The diagram files should be checked into the version control system. Any alternative designs, design decisions, requirements clarifications and notes

are also recorded and written up in the design alternatives section of the Design Package.

Refine the Object Model	Chief Programmer	Required

The Chief Programmer creates a Feature Team Area for the feature(s). This area is either a directory on the file server or a directory on their PC that is backed up to the server by the Chief Programmer as required or utilises work area support in your version control system. The purpose of the Feature Team Area is that work in progress by the feature team can be shared and is visible amongst the feature team but is not visible to the rest of the project.

The Chief Programmer refines the model to add additional classes, methods, attributes and/or to make changes to existing classes, methods or attributes based on the sequence diagram(s) defined for the feature(s). This results in the implementation language source files being updated in the Feature Team Area. The Chief Programmer creates model diagrams in a publishable format. These files should be checked into the version control system and submitted for publishing on the project intranet.

Write Class and Method Prologues	Feature Team	Required

Using the updated implementation language source files from the "Refine the Object Model" task in the shared Feature Team Area, the development owner of each class writes the class and method prologues for each item defined by the feature and sequence diagram(s). This includes parameter types, return types, exceptions and messages. Once each developer has completed this task, the Chief Programmer generates the API documentation using <your tool> and submits it for publication on the project intranet.

Verification

Design Inspection	Feature Team	Required

A design inspection with the feature team members or with other project members is held. The decision to inspect within the feature team or with other project team members is that of the Chief Programmer. On acceptance a to-do list is generated per affected class, and each team member adds their tasks to their calendar task list. The Chief Programmer must also merge changes from the shared Feature Team Area into the change control system.

Exit Criteria

The result of the process is a successfully inspected Design Package. The design package comprises

- A covering memo, or paper, that integrates and describes the design package such that it stands on its own for reviewers.
- The referenced requirements (if any) in the form of documents and all related confirmation memos and supporting documentation.
- The Sequence diagram(s).
- Design alternatives (if any)
- The object model with new/updated classes, methods and attributes.
- The <your tool> generated output for the class and method prologues created or modified by this design.
- Calendar/To-Do task-list entries for action items on affected classes for each team member.

FDD Process #5: Build By Feature

A per-feature activity to produce a completed client-valued function (feature).

Starting with the design package, the development class owners implement the items necessary for their class to support the design for this feature. The code developed is then unit tested and code inspected - the order of which is determined by the Chief Programmer. After a successful code inspection, the code is promoted to the Build.

Entry Criteria

- The Design by Feature process has completed. That is, the design package has successfully been inspected.

Tasks

Implement Classes and Methods	Feature Team	Required

The development class owners implement the items necessary to satisfy the requirements of their class for this feature.

Code Inspection	Feature Team	Required

A code inspection with the feature team members or with other project members is held either before or after the unit test task. The decision to inspect within the feature team or with other project team members is that of the Chief Programmer. The decision to inspect before or after unit test is that of the Chief Programmer.

Unit Test	Feature Team	Required

The development class owner tests their code to ensure all requirements of their class are satisfied. The Chief Programmer determines what feature team-level unit testing is required (if any). That is, if any testing across the classes developed for this feature is required.

Promote to the Build	Chief Programmer, Feature Team	Required

Classes can only be promoted to the build after a successful code inspection. The Chief Programmer tracks the individual classes being promoted, through feedback from the developers, and is the integration point for the entire feature.

Verification

Code Inspection and Unit Test	Chief Programmer, Feature Team	Required

A successful code inspection plus the successful completion of unit test is the verification of the output of this process. The code inspection and unit test tasks are described above.

Exit Criteria

The result of the process is

- Class(es) and/or method(s) that have been successfully code inspected.
- Class(es) that have been promoted to the build.
- The completion of a client-valued function (feature)