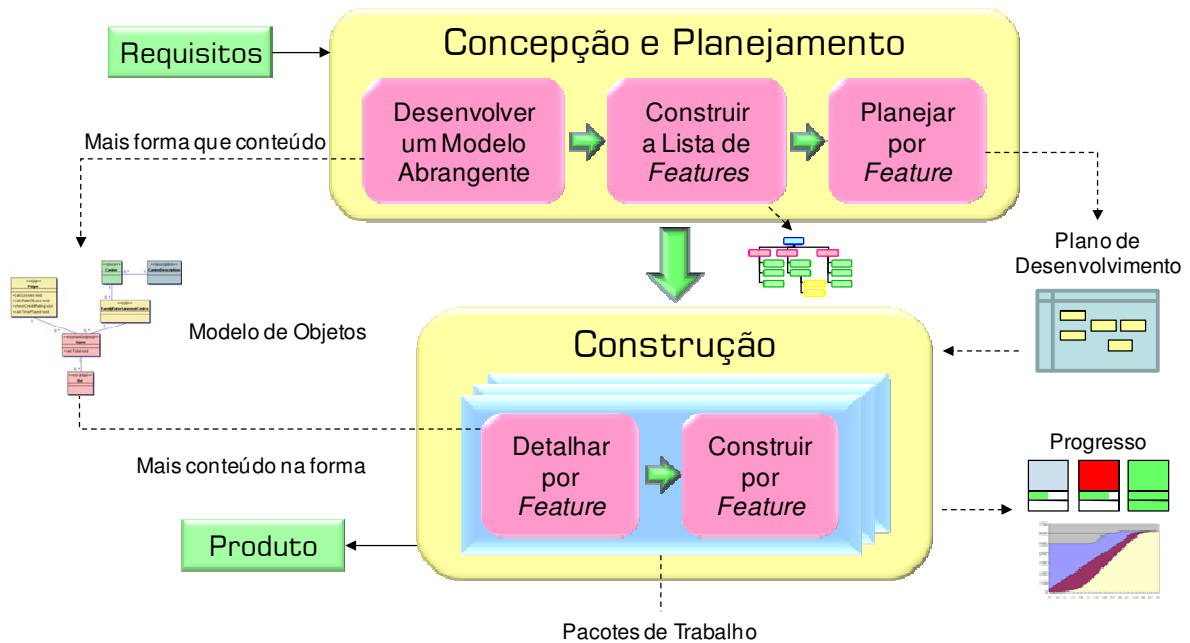


# Feature-Driven Development

## Descrição dos Processos



Tradução da descrição oficial em [www.featuredrivenddevelopment.com](http://www.featuredrivenddevelopment.com)

por Adail Muniz Retamal  
[www.heptagon.com.br](http://www.heptagon.com.br)

V. 1.2 – Maio/2008

## FDD – Processo nº 1: Desenvolver um Modelo Abrangente

É uma atividade inicial que abrange todo o projeto, realizada por membros do domínio do negócio e por desenvolvedores, sob a orientação de um modelador de objetos experiente, no papel de arquiteto líder.

Realiza-se um estudo dirigido sobre o escopo do sistema e seu contexto. Então, são realizados estudos mais detalhados sobre o domínio do negócio para cada área a ser modelada. Após cada estudo dirigido sobre o domínio, pequenos grupos são formados por membros do domínio do negócio sendo estudado e por desenvolvedores, que comporão seus próprios modelos que satisfaçam o domínio em questão. Os pequenos grupos apresentam seus modelos para serem revisados por parceiros e para discussão. Um dos modelos propostos, ou uma combinação dos modelos, é selecionada por consenso, tornando-se, assim, o modelo para aquela área do domínio do negócio. Realiza-se, então, uma combinação do modelo da área do domínio dentro de um modelo abrangente, ajustando a forma do modelo se for necessário.

O modelo de objetos é, então, iterativamente atualizado em seu conteúdo pelo processo nº 4 “Detalhar por Funcionalidade”.

### ***Critérios de Entrada***

- Os especialistas no domínio do negócio, os programadores líderes e o arquiteto líder foram selecionados.

### ***Atividades***

<b>Formar a Equipe de Modelagem</b>	<b>Gerente do Projeto</b>	<b>Obrigatória</b>
-------------------------------------	---------------------------	--------------------

A equipe de modelagem é composta de membros permanentes das áreas do domínio do negócio e de desenvolvimento, especificamente os especialistas no domínio e os programadores líderes. É feito um rodízio com os outros integrantes do projeto através das sessões de modelagem, de modo que todo mundo tenha a chance de participar e ver o processo em ação.

<b>Estudo Dirigido Sobre o Domínio</b>	<b>Equipe de Modelagem</b>	<b>Obrigatória</b>
--	----------------------------	--------------------

Um especialista no domínio do negócio apresenta uma visão geral da área do domínio que será modelada. Essa apresentação deve também incluir informação que estiver relacionada a esta área do domínio, mas não necessariamente uma parte de sua implementação.

<b>Estudar a Documentação</b>	<b>Equipe de Modelagem</b>	<b>Opcional</b>
-------------------------------	----------------------------	-----------------

A equipe estuda os documentos de referência ou de requisitos disponíveis, tais como modelos de objetos, requisitos funcionais (tradicional ou no formato de casos de uso), modelos de dados e guias do usuário.

<b>Desenvolver o Modelo</b>	<b>Eq. Modelagem em Pequenos Grupos</b>	<b>Obrigatória</b>
-----------------------------	---	--------------------

Formando grupos com não mais de três componentes, cada pequeno grupo comporá um modelo que suporte a área do domínio. O arquiteto líder pode propor um modelo base para facilitar o progresso das equipes. Um membro de cada grupo apresenta o modelo proposto por seu grupo para a área do domínio. O arquiteto líder também pode propor outros modelos alternativos. A equipe de modelagem seleciona um modelo proposto ou compõe um modelo pela combinação das idéias propostas nos modelos apresentados.

<b>Refinar o Modelo de Objetos Abrangente</b>	<b>Arquiteto Líder, Equipe de Modelagem</b>	<b>Obrigatória</b>
---	---	--------------------

Freqüentemente, o modelo de objetos abrangente é atualizado com novas formas de modelo produzidas pelas iterações da atividade “Desenvolver o Modelo” descrita acima.

---

### **Verificação**

<b>Avaliação Interna e Externa</b>	<b>Equipe de Modelagem, Negócio</b>	<b>Obrigatória</b>
------------------------------------	-------------------------------------	--------------------

Realiza-se uma auto-avaliação ou uma avaliação interna através da participação ativa dos especialistas no domínio. Quando necessária, uma avaliação externa pode ser feita pedindo-se ao negócio (usuários) que confirme ou esclareça as questões que afetam o modelo.

---

### **Crítérios de Saída**

O resultado do processo é o modelo de objetos:

- Diagramas de classes com foco na forma do modelo, isto é, quais classes estão no domínio, como estão conectadas umas às outras e sob quais restrições;
- Métodos e atributos identificados são colocados nas classes;
- Diagrama(s) de seqüência, se houver;
- Comentários sobre o modelo para registrar o motivo pelo qual uma forma de modelo foi escolhida e/ou quais alternativas foram consideradas.

## FDD – Processo nº 2: Construir a Lista de Funcionalidades

É uma atividade inicial que abrange todo o projeto, para identificar todas as funcionalidades que satisfaçam aos requisitos.

Uma equipe, geralmente composta apenas por programadores líderes do processo nº 1, é formada para decompor funcionalmente o domínio em **áreas**, **atividades de negócio** dentro delas e **passos** dentro de cada atividade de negócio, formando assim a lista categorizada de funcionalidades. A categorização de mais alto nível para a lista de funcionalidades vem da divisão do domínio feita pelos especialistas do domínio no processo nº 1.

### ***Critérios de Entrada***

- Os especialistas no domínio do negócio, os programadores líderes e o arquiteto líder foram selecionados.

### ***Atividades***

<b>Formar a Equipe da Lista de Funcionalidades</b>	<b>Gerente do Projeto, Gerente de Desenvolvimento</b>	<b>Obrigatória</b>
--	---	--------------------

A equipe é composta por programadores líderes da equipe de modelagem do processo nº 1.

<b>Construir a Lista de Funcionalidades</b>	<b>Equipe da Lista de Funcionalidades</b>	<b>Obrigatória</b>
---	---	--------------------

A equipe deve identificar o conjunto de funcionalidades usando o conhecimento adquirido no processo nº 1. Esta é simplesmente uma decomposição funcional nas áreas definidas a partir da divisão do domínio pelos especialistas em cada domínio nos diversos estudos dirigidos realizados no processo nº 1. Ela é decomposta em áreas que englobam atividades de negócio, que são, por sua vez, decompostas em passos (funcionalidades). As funcionalidades são funções granulares, expressas em termos que possuem valor para o cliente, usando o seguinte modelo de nomeação:

<ação> <resultado> <objeto>

Exemplos: calcular o total de uma venda, calcular a quantidade total vendida por um varejista para uma descrição de produto.

As funcionalidades são granulares de acordo com a regra de que uma funcionalidade não levará mais do que duas semanas para ser completada, mas não tão granulares ao nível de “getters” e “setters” (os métodos de acesso aos atributos de uma classe). Duas semanas são um limite superior; a maioria das funcionalidades leva muito menos tempo do que isso. Quando um passo de uma atividade de negócio parece maior do que duas semanas, o passo é quebrado em passos menores, que então tornam-se funcionalidades.

## **Verificação**

<b>Avaliação Interna e Externa</b>	<b>Equipe da Lista, Negócio</b>	<b>Obrigatória</b>
------------------------------------	---------------------------------	--------------------

Realiza-se uma auto-avaliação ou uma avaliação interna através da participação ativa dos membros da equipe de modelagem. Quando necessária, uma avaliação pode ser feita pedindo-se aos especialistas no domínio do negócio da equipe de modelagem ou ao negócio (usuários) que confirmem ou esclareçam as questões que afetam a lista de funcionalidades.

---

## **Crítérios de Saída**

O resultado do processo é a lista de funcionalidades:

- Uma lista de áreas;
- Para cada área, uma lista de atividades de negócio dentro daquela área;
- Para cada passo da atividade de negócio, uma funcionalidade que satisfaça ao passo.

## FDD – Processo nº 3: Planejar por Funcionalidade

É uma atividade inicial que abrange todo o projeto, para produzir o plano de desenvolvimento.

O gerente de projeto, o gerente de desenvolvimento e os programadores líderes planejam a ordem na qual as funcionalidades serão implementadas, baseada nas dependências entre elas, na carga de trabalho da equipe de desenvolvimento e também na complexidade das funcionalidades a serem implementadas. As principais atividades neste processo não são uma seqüência estrita. Como muitas atividades de planejamento, elas são consideradas em conjunto, com refinamentos feitos a partir de uma ou mais atividades e então considerando os outros novamente.

Um cenário típico é levar em conta a seqüência de desenvolvimento, depois levar em conta a atribuição das atividades de negócio aos programadores líderes e, ao fazê-lo, considerar quais das classes principais (apenas) são atribuídas a quais desenvolvedores (lembrar que o programador líder também é um desenvolvedor).

Quando esse equilíbrio for alcançado, e a seqüência de desenvolvimento e a atribuição das atividades de negócio aos programadores líderes estiver essencialmente completada, então a posse das classes estará completada (além das classes principais que já foram consideradas para posse).

### ***Critérios de Entrada***

- O processo “Construir a Lista de Funcionalidades” foi completado.

### ***Atividades***

<b>Formar a Equipe de Planejamento</b>	<b>Gerente do Projeto</b>	<b>Obrigatória</b>
--	---------------------------	--------------------

A equipe de planejamento é composta pelo gerente de desenvolvimento e pelos programadores líderes.

<b>Determinar Seqüência de Desenvolvimento</b>	<b>Equipe de Planejamento</b>	<b>Obrigatória</b>
--	-------------------------------	--------------------

A equipe de planejamento deve atribuir uma data (mês e ano apenas) para o término de cada atividade de negócio. A identificação da atividade de negócio e a data de término (e dessa forma a seqüência de desenvolvimento) é baseada em:

- Dependências entre as funcionalidades em termos de classes envolvidas;
- Distribuição da carga de trabalho entre os proprietários das classes;
- Complexidade das funcionalidades a serem implementadas;
- Adiantamento das atividades de negócio de alto risco ou complexidade;
- Consideração de qualquer marco externo (visível) do projeto, como versões “beta”, demonstrações, pontos de verificação e “todos os produtos” que satisfaçam tais marcos.

<b>Atribuir Atividades de Negócio aos Programadores Líderes</b>	<b>Equipe de Planejamento</b>	<b>Obrigatória</b>
---	-------------------------------	--------------------

A equipe de planejamento deve atribuir programadores líderes como proprietários de atividades de negócio. A atribuição é baseada em:

- Seqüência de desenvolvimento;
- Dependências entre as funcionalidades em termos de classes envolvidas;
- Distribuição da carga de trabalho entre os proprietários das classes (lembrar que os programadores líderes também são proprietários de classes);
- Complexidade das funcionalidades a serem implementadas.

<b>Atribuir Classes aos Desenvolvedores</b>	<b>Equipe de Planejamento</b>	<b>Obrigatória</b>
---	-------------------------------	--------------------

A equipe de planejamento deve atribuir desenvolvedores como proprietários de classes. Os desenvolvedores são proprietários de várias classes. A atribuição das classes aos desenvolvedores é baseada em:

- Distribuição da carga de trabalho entre os desenvolvedores;
- Complexidade das classes;
- Uso das classes (ex. alta utilização);
- Seqüência de desenvolvimento.

---

## **Verificação**

<b>Auto-Avaliação</b>	<b>Equipe de Planejamento</b>	<b>Obrigatória</b>
-----------------------	-------------------------------	--------------------

Como o planejamento é uma atividade de equipe, realiza-se uma auto-avaliação pela participação ativa dos programadores líderes, gerente de desenvolvimento e gerente de projeto.

---

## **Critérios de Saída**

O resultado do processo é o plano de desenvolvimento, consistindo em:

- Atividades de negócio com datas de término (mês e ano);
- Programadores líderes atribuídos a atividades de negócio;
- Áreas com datas de término (mês e ano), derivadas da data do último término de suas respectivas atividades de negócio;
- Lista das classes e seus respectivos desenvolvedores proprietários (a lista de proprietários de classes).

## FDD – Processo nº 4: Detalhar por Funcionalidade

É uma atividade para cada funcionalidade, para produzir o pacote de projeto (*design*) para a funcionalidade.

Certo número de funcionalidades são agendadas para desenvolvimento ao atribuí-las a um programador líder. Ele seleciona as funcionalidades para desenvolvimento a partir de sua “caixa de entrada” de funcionalidades atribuídas. Ele pode escolher diversas funcionalidades que utilizem as mesmas classes (e portanto, desenvolvedores). Operacionalmente, com frequência acontece o caso de “conjuntos” de funcionalidades serem agendados para desenvolvimento de uma vez pelo programador líder. Tal conjunto é chamado de Pacote de Trabalho do programador líder.

O programador líder, então, forma uma equipe de funcionalidades, identificando os proprietários das classes (desenvolvedores) que provavelmente serão envolvidos no desenvolvimento das funcionalidades que ele selecionou. Esta equipe produz o(s) diagrama(s) de seqüência para a(s) funcionalidade(s) atribuída(s). O programador líder, então, refina o modelo de objetos, baseado no conteúdo do(s) diagrama(s) de seqüência. Os desenvolvedores escrevem os prefácios das classes e métodos. Realiza-se uma inspeção no projeto (*design*).

### **Critérios de Entrada**

- O processo de planejamento foi completado.

### **Atividades**

<b>Formar a Equipe de Funcionalidades</b>	<b>Programador Líder</b>	<b>Obrigatória</b>
---	--------------------------	--------------------

O programador líder identifica as classes que provavelmente serão envolvidas no projeto deste conjunto de funcionalidades e, conseqüentemente, atualiza o banco de dados de funcionalidades. Da lista de proprietários de classes, o programador líder identifica os desenvolvedores necessários para formar a equipe de funcionalidades. Como parte deste passo, o programador líder cria um novo pacote de projeto (*design*) para a(s) funcionalidade(s) como parte do Pacote de Trabalho.

<b>Estudo Dirigido do Domínio</b>	<b>Especialista no Domínio</b>	<b>Opcional</b>
-----------------------------------	--------------------------------	-----------------

O especialista no domínio apresenta uma visão geral da área do domínio para a funcionalidade a ser projetada. Essa apresentação deve também incluir informação que estiver relacionada a esta funcionalidade, mas que não seja necessariamente uma parte de sua implementação. Esta é uma atividade opcional, baseada na complexidade da funcionalidade e/ou de suas interações.

<b>Estudar a Documentação de Referência</b>	<b>Equipe de Funcionalidades</b>	<b>Opcional</b>
---	----------------------------------	-----------------

A equipe de funcionalidades estuda o(s) documento(s) de referência para a funcionalidade a ser projetada, todos os memorandos de confirmação, desenhos de telas, especificações de interface com sistemas externos e qualquer outra documentação de suporte. Esta é uma atividade opcional, baseada na complexidade da funcionalidade e/ou de suas interações.

<b>Desenvolver o(s) Diagrama(s) de Seqüência</b>	<b>Equipe de Planejamento</b>	<b>Obrigatória</b>
--	-------------------------------	--------------------

Desenvolver o(s) diagrama(s) de seqüência necessário(s) para a funcionalidade a ser projetada. Os arquivos do(s) diagrama(s) devem ser submetidos ao sistema de controle de versões. Quaisquer projetos (designs) alternativos, decisões de projeto, esclarecimentos de requisitos e comentários também são registrados e descritos na seção de alternativas de projeto (design) do Pacote de Projeto (Design).

<b>Refinar o Modelo de Objetos</b>	<b>Programador Líder</b>	<b>Obrigatória</b>
------------------------------------	--------------------------	--------------------

O programador líder cria uma área para a equipe de funcionalidades para a(s) funcionalidade(s). Esta área pode ser um diretório em um servidor de arquivos ou um diretório em seus próprios computadores, que são copiados para o servidor pelo programador líder quando necessário, ou utiliza-se uma área de trabalho fornecida pelo sistema de controle de versões. O propósito da área da equipe de funcionalidades é para que o trabalho em andamento possa ser compartilhado e estar visível pelos membros da equipe de funcionalidades, mas invisível para o resto do projeto.

O programador líder refina o modelo para adicionar novas classes, métodos, atributos e/ou fazer alterações aos já existentes, baseado no(s) diagrama(s) de seqüência definido(s) para a(s) funcionalidade(s). Isto resulta na atualização dos arquivos fontes da linguagem de implementação na área da equipe de funcionalidades. O programador líder cria diagramas do modelo num formato publicável. Esses arquivos devem ser submetidos ao sistema de controle de versões e publicados na intranet do projeto.

<b>Escrever Prefácios de Classes e Métodos</b>	<b>Equipe de Funcionalidades</b>	<b>Obrigatória</b>
--	----------------------------------	--------------------

Utilizando os arquivos fontes da linguagem de implementação atualizados pela atividade “Refinar o Modelo de Objetos”, que estão na área da equipe de funcionalidades, o proprietário da classes escreve os prefácios de classe e métodos para cada item definido pela funcionalidade e pelo(s) diagrama(s) de seqüência. Isto inclui tipos de parâmetros, tipos de retorno, exceções e mensagens. Uma vez que cada desenvolvedor completou sua tarefa, o programador líder gera a documentação da API usando <sua ferramenta> e a submete para publicação na intranet do projeto.

---

## **Verificação**

<b>Inspeção do Projeto (Design)</b>	<b>Equipe de Funcionalidades</b>	<b>Obrigatória</b>
-------------------------------------	----------------------------------	--------------------

Realiza-se uma inspeção no projeto (design) com os membros da equipe de funcionalidades ou com outros membros do projeto. A decisão de inspecionar com a equipe de funcionalidades ou com outros membros do projeto cabe ao programador líder. Após o aceite, uma lista de tarefas é gerada para cada classe afetada, e cada membro da equipe inclui suas tarefas à sua agenda de tarefas. O programador líder também deve combinar as alterações da área compartilhada pela equipe de funcionalidades de volta ao sistema de controle de versões.

## ***Cr terios de Sa da***

O resultado do processo   um Pacote de Projeto (*Design*) inspecionado com sucesso. O pacote de projeto consiste em:

- Uma capa com coment rios, que completa e descreve o pacote de projeto de tal forma a ser suficiente para futuros revisores;
- Os requisitos referenciados (se houver) na forma de documentos e de todos os memorandos de confirma o relacionados, e documenta o de apoio;
- O(s) diagrama(s) de seq ncia;
- Alternativas de projeto (design) (se houver);
- O modelo de objetos com classes, m todos e atributos novos/atualizados;
- A sa da gerada pela <sua ferramenta> para os pref cios de classes e m todos, criados ou modificados por esse projeto (design);
- Lista de tarefas e agendamentos para itens de a o nas classes afetadas para cada membro da equipe.

## FDD – Processo nº 5: Construir por Funcionalidade

É uma atividade para cada funcionalidade, para produzir uma função com valor para o cliente (funcionalidade).

Começando com o pacote de projeto (design), os proprietários de classes implementam os itens necessários para que suas classes suportem o projeto para esta funcionalidade. O código desenvolvido passa pelo teste de unidade e pela inspeção – a ordem aqui é determinada pelo programador líder. Após passar pela inspeção, o código é promovido à versão atual (build).

### **Critérios de Entrada**

- O processo “Detalhar por Funcionalidade” foi completado, isto é, o pacote de projeto (design) passou com sucesso pela inspeção.

### **Atividades**

<b>Implementar Classes e Métodos</b>	<b>Equipe de Funcionalidades</b>	<b>Obrigatória</b>
--------------------------------------	----------------------------------	--------------------

Os proprietários de classes implementam os itens necessários para satisfazer aos requisitos de suas classes para esta funcionalidade.

<b>Inspeccionar o Código</b>	<b>Equipe de Funcionalidades</b>	<b>Obrigatória</b>
------------------------------	----------------------------------	--------------------

Uma inspeção no código, com membros da equipe de funcionalidades ou outros membros do projeto (a decisão cabe ao programador líder), é realizada antes ou após o teste de unidade (a decisão também cabe ao programador líder).

<b>Teste de Unidade</b>	<b>Equipe de Funcionalidades</b>	<b>Obrigatória</b>
-------------------------	----------------------------------	--------------------

Os proprietários de classes testam seus códigos para certificar que todos os requisitos de suas classes foram satisfeitos. O programador líder determina quais testes de unidade no nível da equipe de funcionalidades são necessários (se houver). Isto é, se algum teste envolvendo as classes desenvolvidas para esta funcionalidade é exigido.

<b>Promover à Versão Atual (Build)</b>	<b>Prog Líder, Equipe de Funcionalidades</b>	<b>Obrigatória</b>
--	--	--------------------

As classes somente podem ser promovidas para a versão atual (*build*) após uma inspeção de código com sucesso. O programador líder monitora as classes sendo promovidas individualmente, através de informações dos desenvolvedores, e é o ponto de integração para a funcionalidade inteira.

## **Verificação**

<b>Inspeção do Código e Teste de Unidade</b>	<b>Prog Líder, Equipe de Funcionalidades</b>	<b>Obrigatória</b>
--	--	--------------------

Uma inspeção de código com sucesso, juntamente com o término dos testes de unidade com sucesso, formam a verificação da saída deste processo. A inspeção do código e o teste de unidades são descritos acima.

---

## **CrITÉrios de Saída**

O resultado do processo é:

- Classes e/ou métodos que passaram na inspeção de código com sucesso;
- Classes que foram promovidas à versão atual (*build*);
- O término de uma função com valor para o cliente (funcionalidade).