

ENGENHARIA DE SOFTWARE

FDD

(Feature Driven Development)

Desenvolvimento Guiado por
Funcionalidades

Prof. Jeangrei Veiga

jeangrei@upf.br

História

O Desenvolvimento guiado a funcionalidades (FDD) foi criado em 1997 num grande projeto de sistema de empréstimos em Java para o Banco United Overseas Bank, em Singapura.

Problema: Após dois anos de consultoria, 3.500 páginas de casos de uso e um modelo de objetos com centenas de classes, foi considerado impossível.

Decisão: União entre a experiência de análise e modelagem orientadas a objetos de **Peter Coad** e o gerenciamento de projetos de **Jeff De Luca**.

Resultado: 15 meses depois da contratação da dupla, 2000 features entregues por uma equipe de 50 pessoas.



Peter Coad



Jeff De Luca

(Os criadores da metodologia FDD)

O que é FDD?

A FDD é uma metodologia ágil para gerenciamento e desenvolvimento de Software, com foco na entrega frequente de “software funcionando” para os clientes e na utilização de boas práticas durante o ciclo de seu desenvolvimento.

Ela combina as melhores práticas do gerenciamento ágil de projetos com uma abordagem completa para Engenharia de Software orientada a objetos, conquistando os três principais públicos de um projeto de software:

✧ Clientes

✧ Gerentes

✧ Desenvolvedores

"O **Ornitorrinco** era originalmente o **projeto de um pato**, mas houve falha na comunicação entre os desenvolvedores e o cliente".



Desenvolvimento ágil

(+)	(-)
Indivíduos e Interações	Processos e Ferramentas
Software Funcionando	Documentação Extensa
Colaboração do Cliente	Negociação de Contrato
Resposta às Mudanças	Seguir um Plano

“Resultados frequentes, tangíveis e funcionais.”

O que são Features?

Features são características ou Funcionalidades que representam algum valor para o cliente e devem ser expressas da seguinte forma:

<action><result><object>

Exemplo:


ação resultado objeto
Calcular o total de uma venda.

Práticas da FDD

- ✧ Modelagem de Objetos do Domínio;
- ✧ Desenvolvimento por features;
- ✧ Posse individual de classe (código);
- ✧ Equipe de features;
- ✧ Inspeções;
- ✧ Builds regulares;
- ✧ Gerenciamento de configuração;
- ✧ Relatório /Visibilidade de Resultados;

Papéis Principais

Especialistas no Domínio

**Gerente de
projeto**

**Gerente de
Desenvolvimento**

**Programadores
Chefes**

**Proprietários das
classes**

Arquiteto Chefe

Da mesma forma é a FDD.

Trata-se de uma metodologia muito objetiva, que promete resultados periódicos rápidos e satisfatórios, sendo estes períodos equivalentes a uma semana ou menos.

Sua objetividade a divide em apenas **duas fases**:

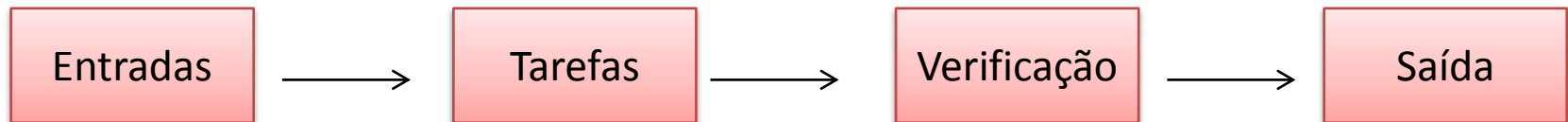
FASES	Descrição	Período
1- Concepção e Planejamento	Pensar antes de fazer	1 a 2 semanas
2- Construção	Fazer de forma iterativa	2 semanas

Processo

Descrição

Cada processo é Descrito em não mais do que duas página de papel tamanho carta, frente e verso.

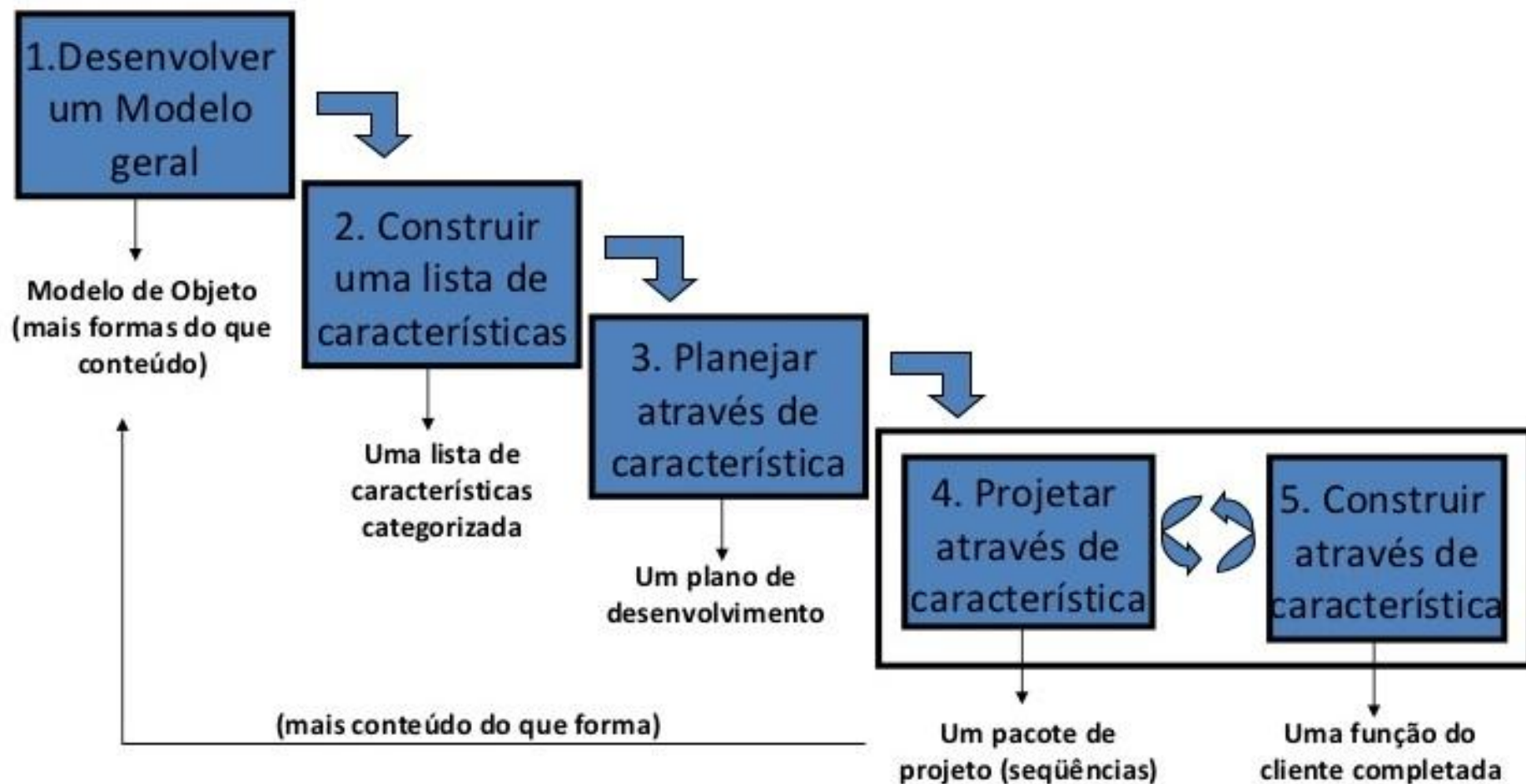
Cada descrição do processo apresenta-se de acordo com a Estrutura:



Processos da FDD

A FDD, classicamente, possui cinco processos.

- ✧DMA- Desenvolver um Modelo Abrangente
- ✧CLF- Construir uma Lista de Funcionalidades
- ✧PPF- Planejar Por Funcionalidade
- ✧DPF- Detalhar por Funcionalidade
- ✧CPF- Construir por Funcionalidade



Processo 1

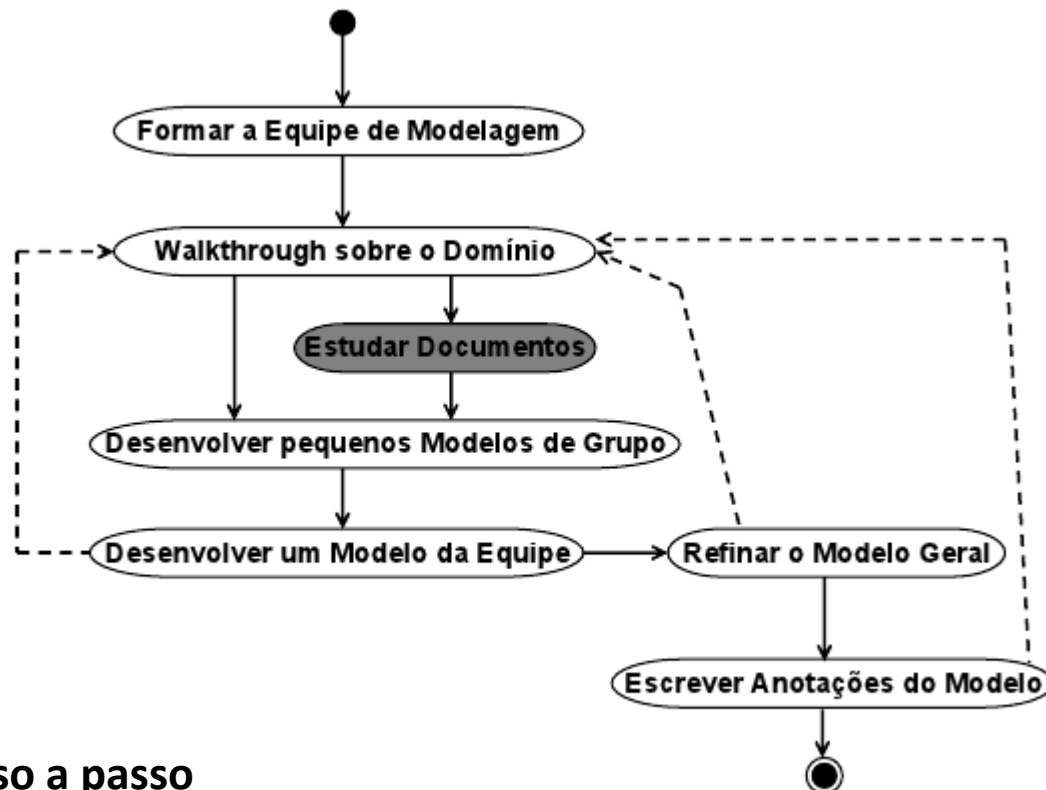
DMA- Desenvolver um Modelo Abrangente

- ✓ Adquirir conhecimento do domínio e desenvolver um modelo geral
- ✓ Estabelecer o “propósito do negócio” do novo sistema.
- ✓ Construção de um “modelo conceitual” do sistema.

Entradas

Especialistas no domínio, programadores e arquitetos chefes são selecionados.

Atividades



*walkthrough – passo a passo

Saídas

- ✓ Modelo Geral do Domínio;
- ✓ Diagrama das Classes principais com alguns métodos e atributos identificados;
- ✓ Diagramas de Sequência de algumas funcionalidades mais complexas (se houver);
- ✓ Comentário sobre o modelo;

Processo 2

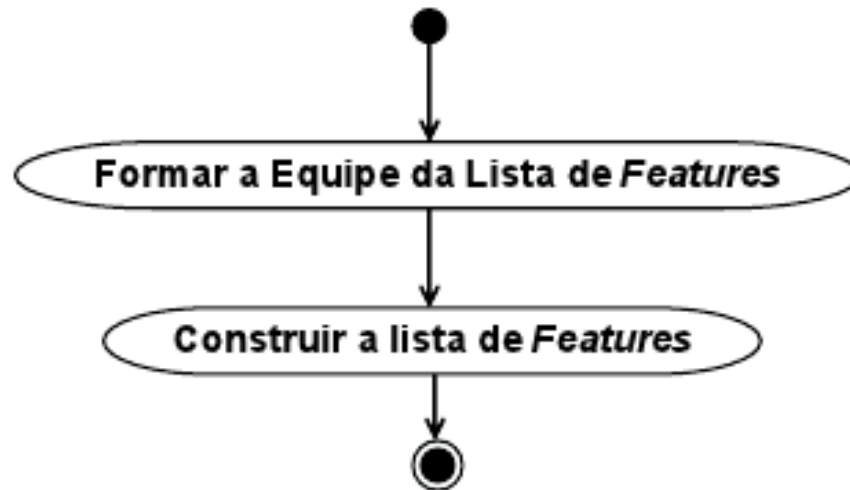
CLF- Construir uma Lista de Funcionalidades

- ✓ O domínio é decomposto até chegar nas funcionalidades;
- ✓ As funcionalidades são agrupadas e categorizadas;
- ✓ As funcionalidades são granuladas até ser necessário menos de 2 semanas para seu desenvolvimento;

Entradas

O Processo 1 ter sido concluído com sucesso.

Atividades



Saídas

- ✓ Uma lista de áreas do domínio identificadas;
- ✓ Para cada área, uma lista de atividades de negócio (conjunto de funcionalidades);
- ✓ Para cada atividade são definidos os passos a serem realizados(Funcionalidades);

Processo 3

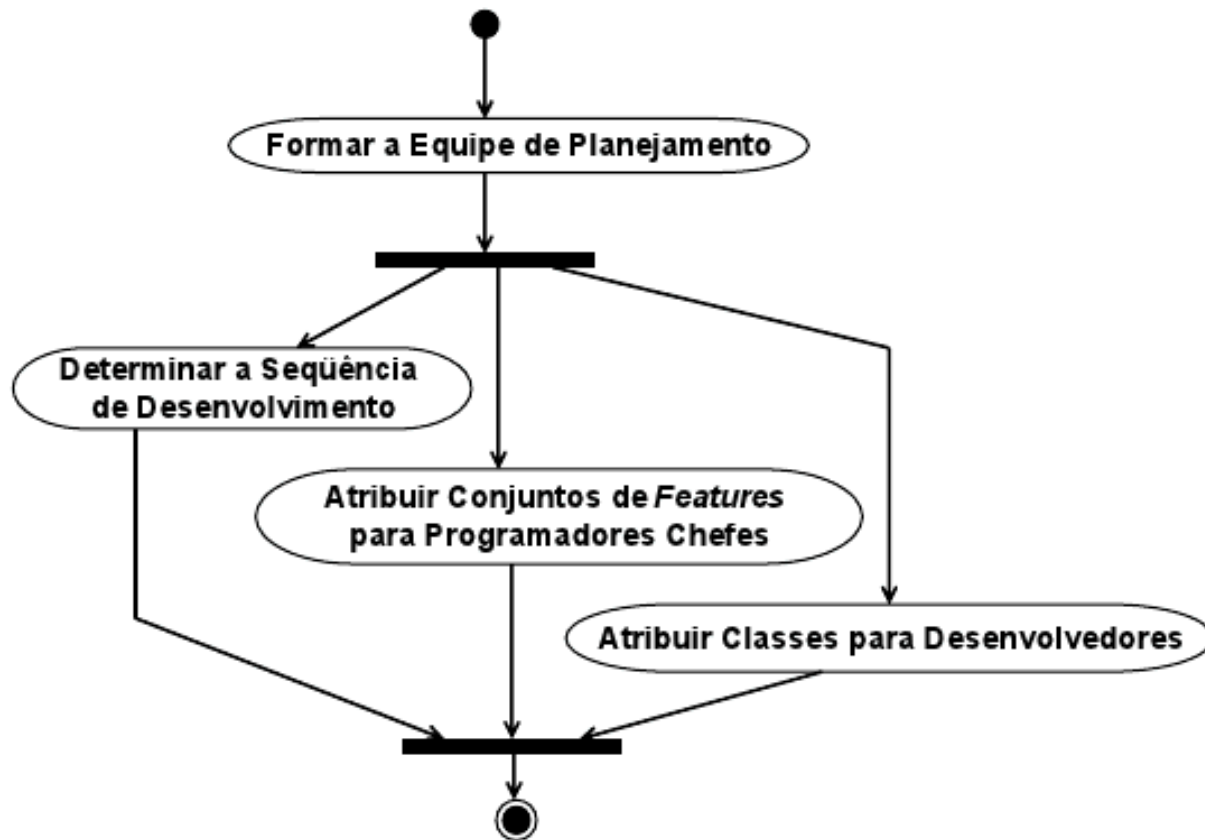
PPF- Planejar Por Funcionalidade

- ✓ Uma data de lançamento é estabelecida para o release inicial;
- ✓ A lista de funcionalidades priorizadas é refinada;
- ✓ O Trabalho técnico é planejado e atribuído;

Entradas

O Processo 2 ter sido concluído com sucesso.

Atividades



Saídas

- ✓ Atividades de negócio com data de término;
- ✓ Programadores chefes atribuídos a atividades de negócio;
- ✓ A lista de classes e seus donos (desenvolvedores);

Processo 4

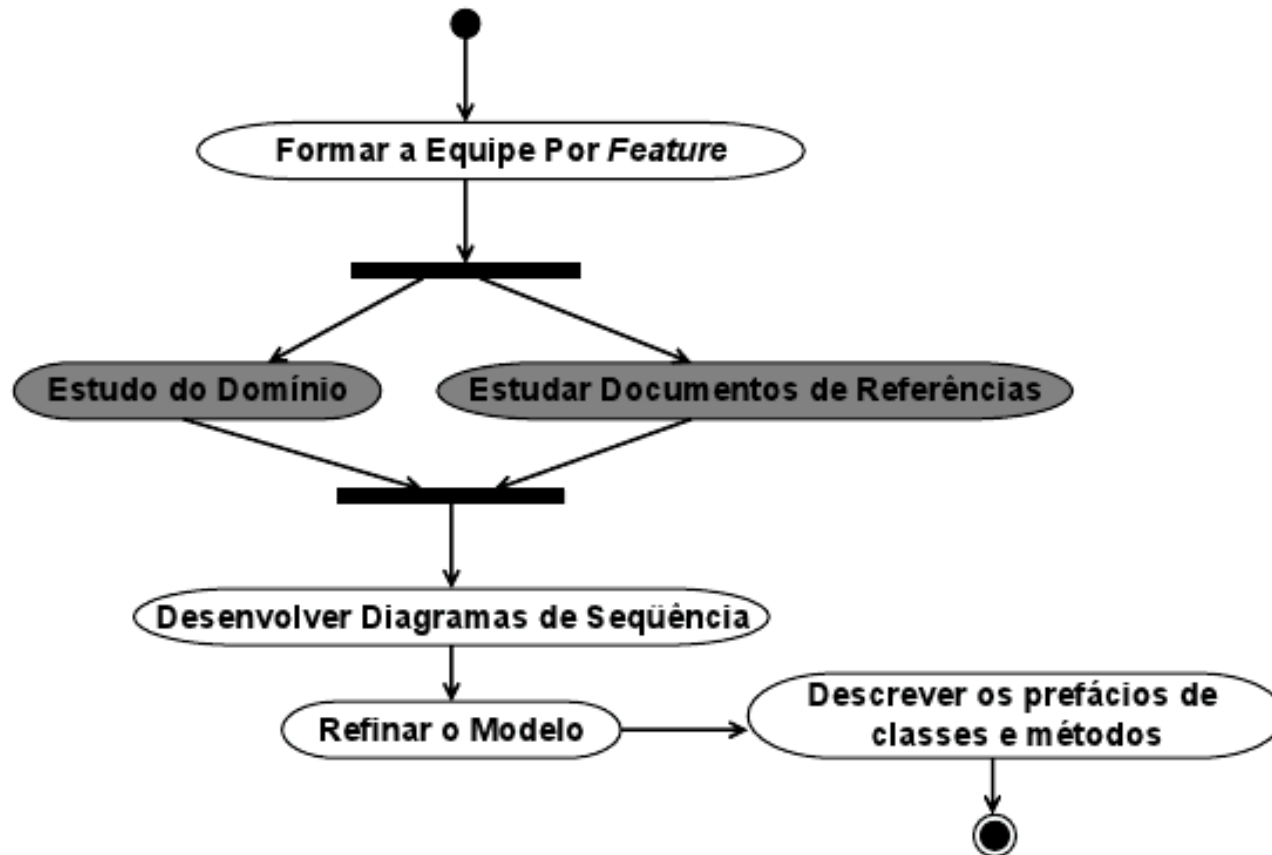
DPF- Detalhar por Funcionalidade

- ✓ Regras e transações são identificadas;
- ✓ O modelo da interface do usuário é esboçado;
- ✓ Diagramas de sequência mais detalhados são produzidos;
- ✓ Especialistas são consultados para descobrir qualquer necessidade específica adicional;

Entradas

O Processo 3 ter sido concluído com sucesso.

Atividades



Saídas

- ✓ Diagramas de sequência;
- ✓ Projetos alternativos (caso exista);
- ✓ O modelo de objeto com classes, métodos e atributos novos ou atualizados ;
- ✓ Lista de tarefas(calendário/o que fazer)

Processo 5

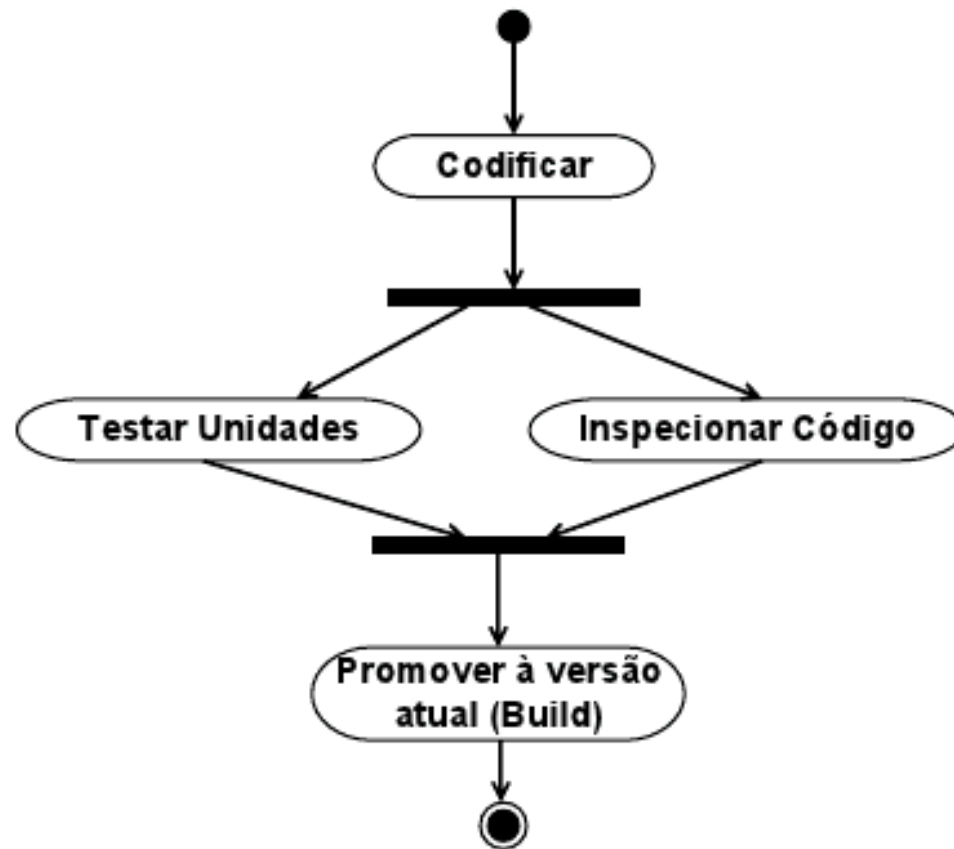
CPF- Construir por Funcionalidade

- ✓ As funcionalidades são construídas implementando todas as classes e métodos necessários;
- ✓ Testes de unidades;
- ✓ As funcionalidades são inseridas no build quando o teste resulta em sucesso;

Entradas

O processo anterior (Processo 4) ter sido concluído com sucesso.

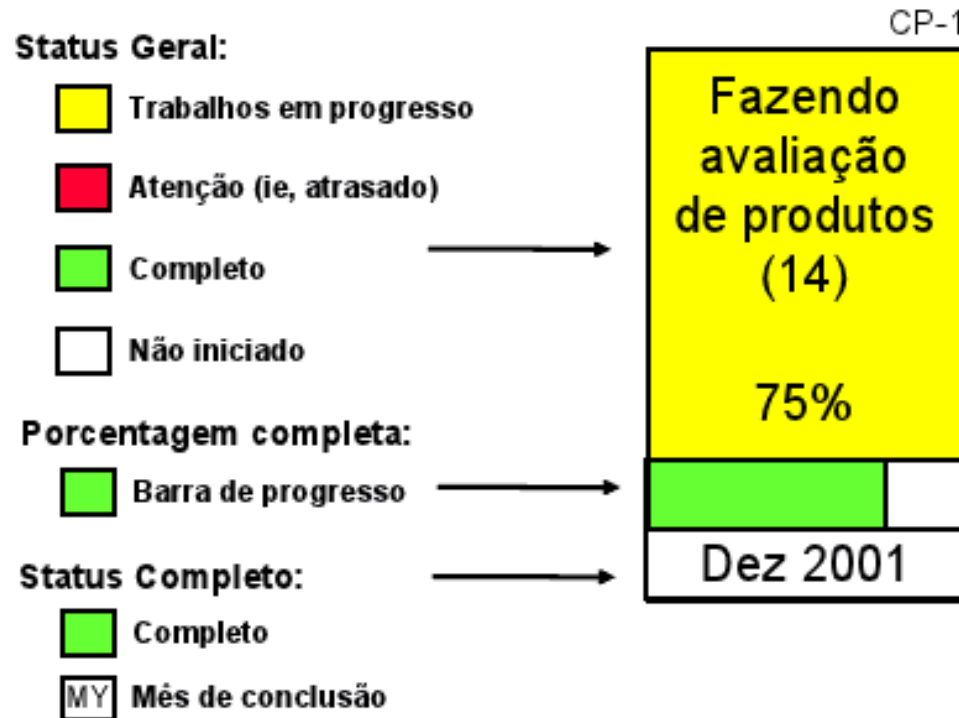
Atividades



Saídas

- ✓ Classes e/ou métodos que passaram na inspeção do código com sucesso;
- ✓ Classes inseridas no build;
- ✓ A conclusão da funcionalidade do cliente ;

Demonstrando Resultados



Exemplo:

Conjunto de *Features*: Fazendo avaliação de produtos – Trabalho em progresso

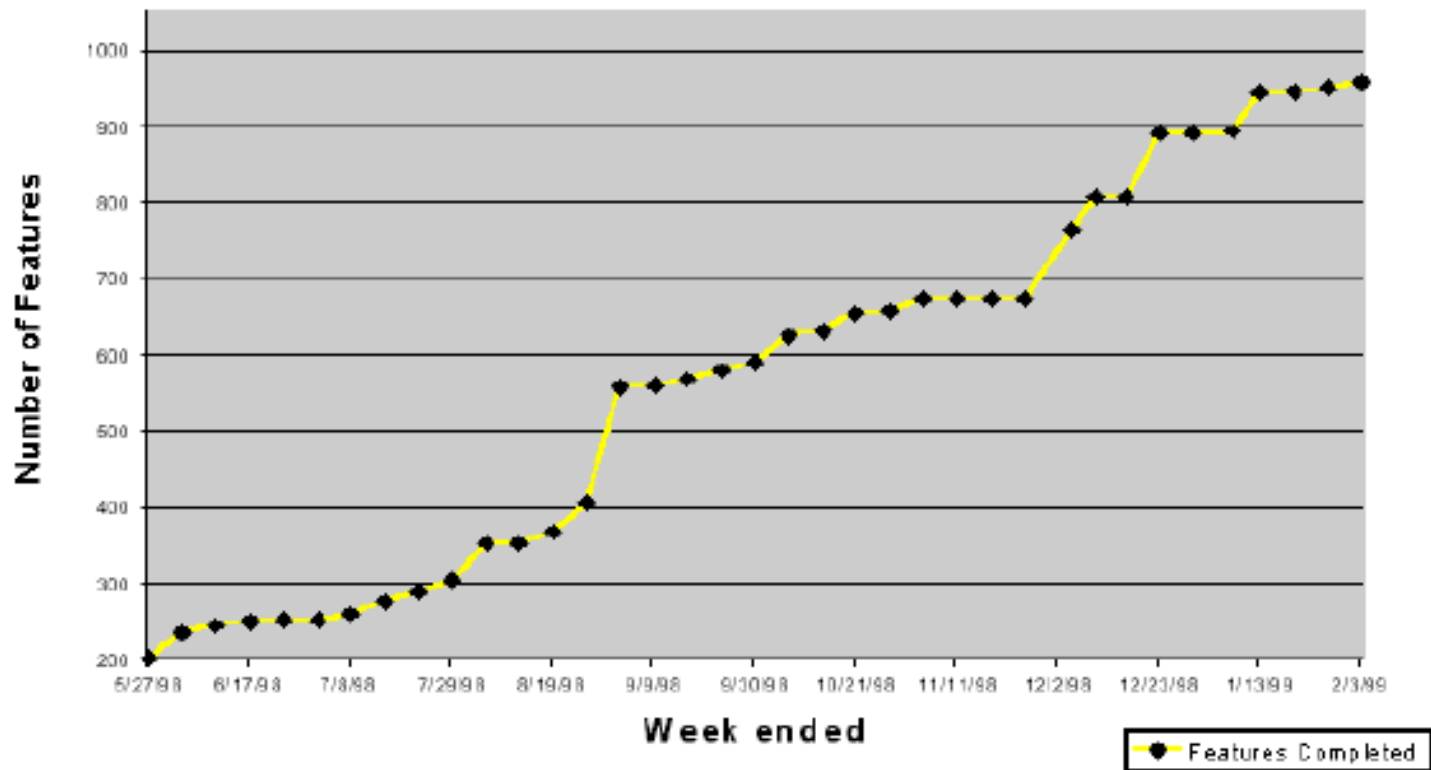
CP-1 é o programador chefe inicial

(14) esse conjunto de *Features* possui 14 características

Conjunto de *Features*/ está 75% completado

A conclusão é para dezembro de 2001

CLS - PD Completed Features on a Weekly basis



Os seis marcos do FDD

Projetar pelas características			Construir pelas características		
Análise do domínio	Projeto	Inspeção do projeto	Código	Inspeção do código	Geração de build
1%	40%	3%	45%	10%	1%

Modelo disponibilizado pela empresa de desenvolvimento de software **Sys Evolution** que utiliza a FDD como metodologia de gerenciamento e desenvolvimento de seus softwares.

