

FDD

Caio Sanches Bentes
caio@ieee.org

STSI 4 - Semana de Tecnologia e Segurança da Informação

28 de Março de 2019

Agenda

1 FDD

EXPERIÊNCIA

EXPERIÊNCIA

- GTSET CTIC-UFPA DESENVOLVEDOR DE SOFTWARE
Julho 2014 – Dezembro 2016 — Belém, PA
- LABES - UFPA — ENGENHEIRO DE SOFTWARE - INICIAÇÃO CIENTÍFICA
Abril 2017 – Agosto 2017 — Belém, PA
- DATACENTER - UFPA — SYSADMIN JUNIOR
Março 2017 – Janeiro 2018 — Belém, PA
- LASSE - UFPA — INTERNO
Novembro 2017 – Março 2018 — Belém, PA
- DATACENTER - FADESP — SYSADMIN JUNIOR
Janeiro 2018 – Maio 2018 — Belém, PA
- SOLUS TECNOLOGIA — ANALISTA DE INFRAESTRUTURA JUNIOR
Maio 2018 - Atualmente — Belém, PA

DOCÊNCIA

- UNIVERSIDADE FEDERAL DO PARÁ — MONITOR DE PROCESSAMENTO DIGITAL DE SINAIS
Maio 2016 – Agosto 2017 — Belém, PA

Definição

FDD – Feature Driven Development

(Desenvolvimento Guiado por Funcionalidades): É uma metodologia ágil para gerenciamento e desenvolvimento de software. Ela combina as melhores práticas do gerenciamento ágil de Projetos com uma abordagem completa para Engenharia de Software orientada por objetos.

O que é Feature?

FDD – Feature Driven Development

Funcionalidade: É uma funcionalidade para o detalhamento e uma característica pequena para ser implementada, no máximo em um iteração, oferecendo assim o valor ao cliente

< *ação* > < *resultado* > < *objeto* >

História da FDD

História da FDD

- O FDD foi criado em 1997 num grande projeto em Java para o United Overseas Bank, em Cingapura.
- Nasceu a partir da experiência de análise e modelagem orientadas por objetos de Peter Jeff de Luca Coad e de gerenciamento de projetos por Jeff de Luca.
- Foi inicialmente publicada em 1999, no capítulo 6 do livro “Java Modeling in Color with UML”, de Peter Coad, Eric Lefebvre e Jeff de Luca. Peter Coad
- Seu lema é: “Resultados frequentes, tangíveis e funcionais.”

Características

A FDD chama a atenção por algumas características peculiares:

- Resultados úteis a cada duas semanas ou menos bem pequenos de funcionalidade valorizada pelo cliente, chamados "Features" não existem restrições quanto à complexidade do sistema e tamanho da equipe
- Planejamento detalhado e guia para medição
- Rastreabilidade e relatórios com precisão
- Monitoramento detalhado dentro do projeto, com resumos de alto nível para clientes e gerentes, tudo em termos de negócio
- Fornece uma forma de saber, dentro dos primeiros 10 % de um projeto, se o plano e a estimativa são sólidos



Laboratório de Processamento de Sinais

Padrões do FDD

Segundo De Luca todas as fases do FDD devem seguir o padrão “ETVX” :

- Entry – Entrada: define e especifica critérios de entrada para as fases do FDD;
- Task – Tarefa: é composto por uma lista de tarefas a ser realizada a cada uma das fases;
- Verification – Verificação: especifica tipos de avaliações e inspeções de projeto e códigos “testes”;
- Exit – Saída: especifica os critérios de saída ou seja os critérios de “pronto” da fase;

Práticas do FDD

Práticas do FDD

- Modelagem dos objetos de domínio;
- Desenvolvendo através de funcionalidades;
- Propriedade individual das classes;
- Equipes de funcionalidades; Inspeções;
- Construções regulares; Administração de configuração;
- Relatórios de resultados;

Práticas do FDD

Modelagem dos objetos de Domínio

- Construção de diagramas de classes UML (Unified Modeling Language) que descrevem os objetos relevantes dentro do domínio do problema, bem como os relacionamentos entre eles. Para complementar os diagramas de classe UML, são desenvolvidos diagramas de sequência UML que descrevem explicitamente como os objetos interagem para cumprir suas responsabilidades.

Práticas do FDD

Desenvolvendo através de funcionalidades

- É feita a identificação das funcionalidades do sistema (definidas pelo cliente). Após isso, inicia-se o projeto e a construção de cada uma delas. Uma vez identificadas, as funcionalidades serão utilizadas para guiar o desenvolvimento no FDD, tendo como objetivo mostrar o progresso através da implementação das mesmas.
- A execução das funcionalidades, ou conjunto delas, não deve exceder de duas semanas.

Práticas do FDD

Propriedade individual das classes

- ada classe ou conjunto de classes é de responsabilidade de um indivíduo.
- Isso pode ser uma vantagem e uma grande desvantagem em alguns pontos de vista, pois a saída do programador proprietário da classe pode gerar perda de tempo no projeto.

Práticas do FDD

Equipes de funcionalidades

- A prática da propriedade individual da classe atribui classes a desenvolvedores específicos. Contudo, sabe-se que o desenvolvimento deve ser por funcionalidade. Por isso, são definidas equipes, com seus respectivos desenvolvedores líderes, onde os componentes possuem as propriedades das classes, e é atribuído a eles um conjunto de funcionalidades.
- A equipe de funcionalidades deve ter no mínimo 3 e no máximo 6 programadores envolvidos.

Práticas do FDD

Inspeções

- Devem ser feitas durante e ao final de cada iteração, para assegurar a qualidade do projeto e do código. O objetivo principal das inspeções é a detecção de defeitos. É uma ferramenta para eliminação de erros e uma grande oportunidade de aprendizado.

Práticas do FDD

Construções regulares

- Devem ocorrer durante as iterações, na execução de um conjunto de funcionalidades, para detectar, prematuramente, erros de integração. Uma construção regular assegura também que haja sempre um sistema atual e executável para ser apresentado ao cliente.

Práticas do FDD

Administração de configuração

- Utilização de um sistema de controle de versões para datar e manter um histórico das alterações feitas em cada classe. Bem como, no que se refere aos requisitos, análise e o projeto de modo que facilite a visualização das modificações feitas.

Práticas do FDD

Relatórios de resultados

- Relatórios de resultados O FDD sugere que todos os resultados ocorridos durante o projeto sejam disseminados para todos os membros da equipe e clientes.

Papéis

Papéis

- Principais
- Apoio
- Adicionais

Papéis principais

Papéis principais

- Gerente de projeto
- Arquiteto chefe
- Gerente de desenvolvimento
- Programador chefe
- Proprietário de classe
- Especialista do domínio

Papéis principais

Gerente de projeto

- Responsável financeiro e administrativo do projeto. Uma de suas responsabilidades é gerenciar a viabilidade do projeto oferecendo todas as condições necessárias à equipe para o desenvolvimento do trabalho. No FDD, a “última palavra” é dada por ele.

Papéis principais

Arquiteto chefe

- Elabora o projeto geral do software a ser desenvolvido, tomando decisões finais em relação ao projeto técnico. Essa função poderá ser dividida entre o Projetista de Domínio e o Projetista Técnico.

Papéis principais

Gerente de desenvolvimento

- Responsável por gerenciar as atividades diárias do projeto resolvendo problemas que poderão ocorrer com a equipe. Pode combinar as atividades desenvolvidas pelo Arquiteto Principal e Gerente de Projeto.

Papéis principais

Proprietário de classe

- Geralmente é o programador com maior experiência dentro da equipe, participando na análise dos requisitos e no projeto do software. Considerado como um dos papéis mais importantes no projeto FDD. Atua principalmente nas duas últimas etapas do processo.

Papéis principais

Proprietário de classe

- Subordinado do Programador Chefe, tendo como tarefas projetar, codificar, testar e documentar. Responsável pelo desenvolvimento das classes atribuídas a ele.

Papéis principais

Especialista do domínio

- Pode ser um usuário, analista de negócios, cliente ou qualquer pessoa que conheça bem o domínio do problema. Sua tarefa é informar as funcionalidades que deverão ser atendidas pelo software e entender como os requisitos estão sendo desenvolvidos.

Papéis de apoio

Gerente do domínio

- Gerente de versão Especialista (guru) de linguagem
Coordenador de construção “Ferramenteiro” (toolsmith)
Administrador de sistema

Papéis de apoio

Gerente do domínio

- Conduz os peritos de domínio a resolver as diferenças de opinião relativa aos requisitos do sistema.

Papéis de apoio

Gerente de versão

- Responsável por controlar o progresso no desenvolvimento através de constantes revisões em conjunto com o Programador Chefe. Informa suas atividades ao Gerente de Projeto.

Papéis de apoio

Especialista (guru) de linguagem

- Membro da equipe responsável por possuir um conhecimento completo de uma linguagem de programação específica ou tecnologia. Este papel é particularmente importante quando é usada uma nova tecnologia.

Papéis de apoio

Coordenador de construção

- Pessoa responsável pelas tarefas de administração do sistema de controle de versão e a publicação da documentação, durante a atividade de construção.

Papéis de apoio

“Ferramenteiro” (toolsmith)

- Responsável por construir ferramentas de suporte para o desenvolvimento, teste e conversão de dados no projeto. Também pode trabalhar com modelagem e manutenção de bancos de dados e websites para propósitos específicos do projeto.

Papéis de apoio

Administrador de sistema

- Possui a tarefa de configurar, administrar e diagnosticar os servidores, estações de trabalho e desenvolvimento e testar os ambientes usados pela equipe do projeto.

Papéis adicionais

Papéis adicionais

- Testador
- Desenvolvedores
- Escritor técnico

Papéis adicionais

Testador

- Verificam se o sistema que está sendo produzido satisfará os requisitos do cliente.

Desenvolvedor

- Responsáveis por converter dados existentes ao formato requerido pelo novo sistema e participar no desenvolvimento de novos lançamentos

FDD

=

Papéis adicionais

Escritor técnico

- Responsável pela documentação de usuário.

Time

Time

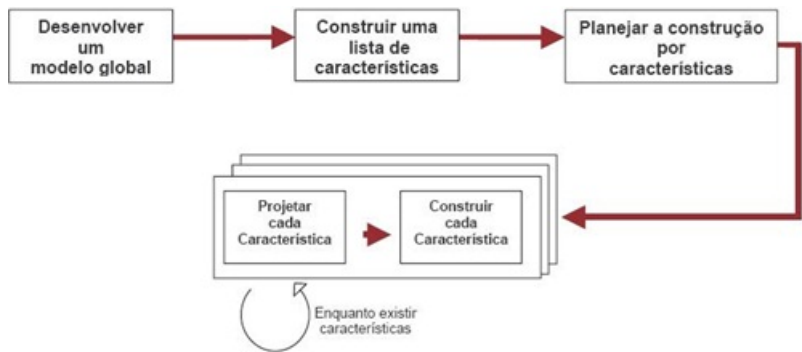
- Formados dinamicamente: única forma de desenvolver por feature (funcionalidade) e manter a posse do código;
- Sob a coordenação de um programador chefe;
- Múltiplas mentes projetando;
- Membros são os Donos de Classes relevantes;
- Enfatiza o trabalho em equipe;

Fases do FDD

Fases do FDD

- A FDD é uma metodologia muito objetiva que possui cinco fases e essas podem ser divididas em duas etapas:
 - **Concepção** Planejamento: Pensar no modelo, criar uma lista de características e planejar através delas. Essa fase é executada apenas uma vez e dura de uma a duas semanas.
 - **Construção**: Desenvolvimento iterativo e incremental durante um período de tempo de no máximo 2 semanas

Fases do FDD



Concepção Planejamento

Concepção Planejamento

- Desenvolver um modelo abrangente
- Construir a lista de funcionalidades
- Planejar por funcionalidade

Desenvolver modelo abrangente

Desenvolver modelo abrangente

- Formar o time de modelagem.
- Conduzir o Domain Walkthrough.
- Estudar documentação.
- Desenvolver modelos de pequenos grupos.
- Desenvolver modelo da equipe.
- Refinar o Modelo Abrangente.
- Escrever Notas.

Desenvolver modelo abrangente

O Modelo de Domínio Criado Será decomposto por 3 camadas:

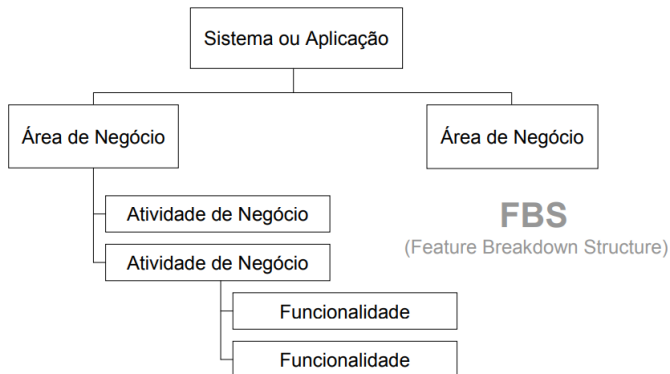
- Área de Negócio.
- Atividade de Negócio.
- Automatização da Atividade (Funcionalidade).

Construir a lista de funcionalidades

Construir a lista de funcionalidades

- Construção de uma lista de funcionalidades importantes para o cliente onde a lista representara o produto final a ser desenvolvido, podendo ser necessário a inclusão de novas funcionalidades no modelo de domínio a cada iteração.

Construir a lista de funcionalidades



Planejar por funcionalidade

Planejar por funcionalidade

- Determinar a seqüência do desenvolvimento .
- Atribuir atividades de Negócio aos Programadores-chefes.
- Atribuir Classes aos Desenvolvedores.

Construção

Construção

- Detalhar por funcionalidade
- Construir por funcionalidade

Detalhar por funcionalidade

Detalhar por funcionalidade

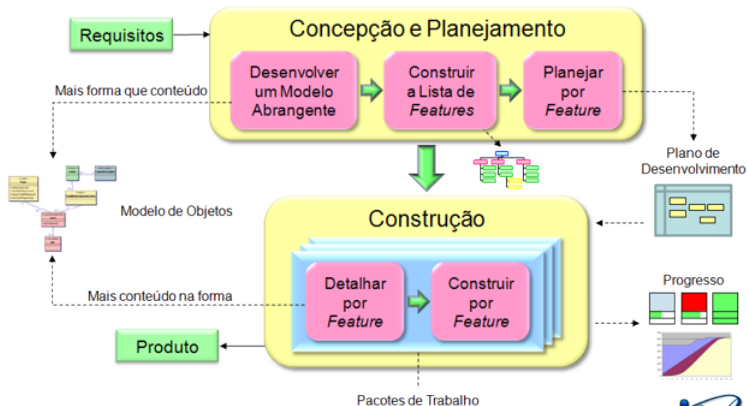
- Já dentro de uma iteração de construção, a equipe detalha os requisitos e outros artefatos para a codificação de cada funcionalidade, incluindo os testes. O projeto para as funcionalidades é inspecionado e o modelo abrangente é atualizado. O resultado é o modelo de domínio mais detalhado e os esqueletos de código como declaração de métodos, tipagem, atributos e parâmetros prontos para serem preenchidos.

Construir por funcionalidade

Construir por funcionalidade

- Cada esqueleto de código é preenchido, testado e inspecionado de acordo com o modelo abrangente. O resultado é um incremento do produto integrado ao repositório principal do código após ter sido devidamente testado por um outro membro da equipe se necessário, com qualidade e potencial para ser usado pelo cliente.

Processo



FDD x Scrum

FDD x Scrum

- Eles são compatíveis?
- Qual é o papel de cada um no desenvolvimento de software?
- Eles são complementares?

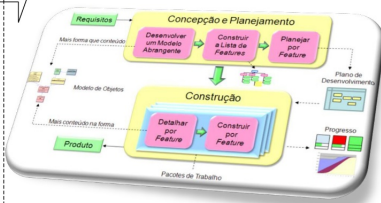
Combinando Scrum x FDD

Gerenciamento (SCRUM) e Engenharia de Software (FDD, XP)



SCRUM

Gestão de Projeto



FDD

Engenharia de software

Vantagens

Vantagens

- Recomendado para qualquer tipo de desenvolvimento;
- Foco em “características de valor para o cliente”;
- FDD prioriza aquilo que o cliente prioriza;
- FDD possui requisitos mais formais;
- Seus princípios e práticas proporcionam um equilíbrio entre as filosofias tradicionais e as mais extremas, proporcionando uma transição mais suave para organizações mais conservadoras;

Desvantagens

Desvantagens

- Ainda existem questionamento sobre a eficácia / aplicabilidade de FDD;
- Controvérsias sobre o tamanho mínimo de um time FDD;

Conclusão

Conclusão

- É um método ágil e altamente adaptativo;
- Oferece vantagens dos métodos pesados;
- Oferece vantagens dos métodos extremamente ágeis;
- É orientada às necessidades dos clientes, gerentes e desenvolvedores;

Referencias

Referencias

- Feature Driven Development; 2007; Setti Rodrigo, Gameiro Lucas, Boscariol Lendro, Leal Flavio
<http://www.featuredrivendevelopment.com/>
- Modelagem da Interação do usuário com o sistema em métodos ágeis; 2009; Cecilia Giuffra Palomino
- FDD Numa Casca de banana, Um guia de rápido aprendizado para a Feature Driven Development; Mar 2007; Alexandre Magno Figueiredo

Perguntas?

Perguntas?

FIM



Caio Sanches Bentes
caio@ieee.org