

Feature Driven Development



Desenvolvimento dirigido à funcionalidades

Jonathas Corrêa, Wanner Menezes e Fábio D.

Introdução

- ❖ Antes de termos os manifestos ágeis existiam alguns métodos ágeis e o FDD esta entre eles.
- ❖ Foi criado em Singapura por Jeff de Lucca entre os anos de 1997 e 1999 com base no método Coad (criado por Peter Coad entre 1980 e 1990).

O que é feature?

- ❖ No contexto de FDD, uma feature pode ser descrita como uma função da necessidade do cliente que pode ser implementada em uma semana ou menos
- ❖ Funcionalidade é uma função com um valor para o cliente

Objetivo

- ❖ O objetivo do FDD é o desenvolvimento por funcionalidade ou seja ele procura algum requisito funcional do sistema.

Praticidades

- ❖ Prático pra trabalhar com projetos iniciais
- ❖ Prático para trabalhar com codificações existentes

Vantagens

- ❖ Maior facilidade de achar erros e ambiguidades
- ❖ Organização de forma hierárquica
- ❖ Menor custo humano já que cada feature pode ser desenvolvida de maneira independente
- ❖ Menor custo de tempo para achar erros

Desvantagens

- ❖ Questionamento sobre a efetividade do FDD
- ❖ Não existe consenso do tamanho que cada feature deve ter

Processos básicos

- ❖ O FDD possui cinco processos básicos
- ❖ Desenvolvimento de modelo abrangente
- ❖ Construção de lista de funcionalidades
- ❖ Planejar por funcionalidade
- ❖ Detalhe por funcionalidade
- ❖ Construção por funcionalidade

Desenvolvimento de modelo abrangente (Análise orientada por objeto)

- ❖ Escopo do sistema e contexto
- ❖ Formação dos grupos
- ❖ Modelo de objetos (diagrama)

Construção de lista de funcionalidades (decomposição funcional)

- ❖ Funcionalidades que satisfaçam os requisitos
- ❖ Decomposição do domínio (áreas e atividades de negócio)
- ❖ Lista de áreas, atividades para cada área e passo de cada funcionalidade

Planejar por funcionalidade (Planejamento incremental)

- ❖ Abrange o projeto para plano de desenvolvimento
- ❖ Ordem das funcionalidades
- ❖ Atividades datadas
- ❖ Lista de desenvolvedores proprietários

Detalhe por funcionalidade (Desenho orientado a objeto)

- ❖ Pacote de projeto (Design)
- ❖ Alternativas de projeto
- ❖ Inspeção de diagramas
- ❖ Modelo de projetos atualizados
- ❖ Capa descritiva
- ❖ Lista de tarefas
- ❖ Diagramas de sequência

Construção por funcionalidade (Programação e teste orientado a objetos)

- ❖ Atividade para funcionalidade
- ❖ Produzir a funcionalidade (função com valor para o cliente)
- ❖ Design: itens para que as classes suportem o projeto para a funcionalidade, inspeção do projeto
- ❖ Término da funcionalidade

Semelhanças - SCRUM

- ❖ Colaborativo
- ❖ Comunicação otimizada
- ❖ Desenvolvido e testado devidamente à cada pequena iteração
- ❖ O progresso pode ser acompanhado em diferentes granularidades
- ❖ Ênfase na produção de componentes de qualidade

Diferenças

SCRUM

- ❖ Não se aplica nenhuma prática da engenharia, embora partes do XP são frequentemente adotados.
- ❖ Concentra-se em produzir fatias verticais de funcionalidade para o proprietário do produto.
- ❖ Loops de feedback mais curtos
- ❖ Equipes auto-organizadas
- ❖ Propriedade compartilhada do código

Diferenças

FDD

- ❖ Práticas de engenharia específicos, ou seja, inspeções de design/código, testes
- ❖ Dirigido ao domínio
- ❖ Ciclo de feedback maior
- ❖ Possui papéis definidos para cada área (ex.: Gerente de Projeto, Arquiteto Chefe, Gerente de Desenvolvimento, etc.)
- ❖ Propriedade do código por setor

Quando escolher FDD?

- ❖ Estiver disposto a trocar um pouco da agilidade para entregar modelos bem definidos e escaláveis
- ❖ Time possui habilidades sólidas de modelagem
- ❖ A maioria dos requerimentos são bem conhecidos e estáveis

Quando escolher Scrum?

- ❖ Requerimentos são dinâmicos e emergentes
- ❖ Autonomia para as equipes
- ❖ Precisa mais de um framework de gerenciamento do que práticas de engenharia.
- ❖ Melhor para administrar riscos

Divisão de tempo

- ❖ Levantamento do domínio da aplicação = 1%
- ❖ Projeto = 40%
- ❖ Inspeção do projeto = 3%
- ❖ Desenvolvimento = 45%
- ❖ Inspeção do código = 10%
- ❖ Integração = 1%.

Referências

- ❖ <http://chrisjasonkelly.blogspot.com.br/2011/08/comparison-of-fdd-and-scrum.html>
- ❖ <http://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971>
- ❖ <https://pt.wikipedia.org/wiki/Granularidade>
- ❖ <http://simpleprogrammer.com/2011/11/21/understanding-the-vertical-slice/>
- ❖ <https://dzone.com/articles/introduction-feature-driven>
- ❖ <http://www.slideshare.net/manoelp/gesto-gil-de-projetos-com-scrum-e-fdd-manoel-pimentel-presentation>