

PROCESSOS DE SOFTWARE

Aula 03 - Introdução a Modelos de Desenvolvimento de Software

Prof. Fabricio Freire

METAS DE APRENDIZAGEM

1. **Compreender** os conceitos do Processo Unificado (UP) e suas ferramentas
2. **Identificar** as características e práticas da metodologia XP
3. **Aplicar** os conceitos da metodologia FDD em projetos
4. **Comparar** as diferentes abordagens metodológicas
5. **Selecionar** a metodologia mais adequada para cada contexto

Por que estudar Modelos de Desenvolvimento?

Evolução das Necessidades

- Projetos cada vez mais complexos
- Requisitos que mudam frequentemente
- Necessidade de entregas mais rápidas
- Qualidade e satisfação do cliente

O Desafio

Como gerenciar projetos de software garantindo qualidade, prazo e custo?

O que é o UP?

Processo Unificado (Unified Process) é uma metodologia de desenvolvimento de software que:

- Gerencia projetos de forma estruturada
- Garante qualidade do produto
- Atende prazos e custos estabelecidos
- Utiliza UML como ferramenta de especificação
- Pode ser combinado com outras metodologias (XP, Scrum)

Característica Principal

É iterativo e incremental - o sistema evolui em ciclos

As 4 Fases do UP

Concepção (Inception)

- Definição do escopo do projeto
- Identificação dos riscos principais
- Estimativa de recursos necessários

Elaboração (Elaboration)

- Análise detalhada dos requisitos
- Arquitetura base do sistema
- Planejamento do projeto

Construção (Construction)

- Desenvolvimento do software
- Implementação das funcionalidades
- Testes unitários e de integração

Transição (Transition)

- Implantação do sistema
- Treinamento dos usuários
- Suporte inicial

Lembre-se: Concepção → Elaboração → Construção
→ Transição

Princípios Fundamentais

Orientado a Casos de Uso

Sistema desenvolvido baseado nas necessidades dos usuários

Centrado na Arquitetura

Arquitetura sólida desde o início

Iterativo e Incremental

Desenvolvimento em ciclos

Sistema evolui gradualmente

Tratamento de Riscos

Riscos críticos são abordados primeiro

O que é XP?

Extreme Programming é uma metodologia ágil que:

- Foca no desenvolvimento rápido
- Lida bem com mudanças frequentes de requisitos
- Trabalha com equipes pequenas (até 12 desenvolvedores)
- Utiliza programação orientada a objetos
- Prioriza a satisfação do cliente

Objetivo Principal

Atender necessidades dos clientes com **qualidade, rapidez e simplicidade**

Os 5 Valores Fundamentais

Comunicação

Comunicação constante entre equipe e cliente

Simplicidade

Desenvolver apenas o necessário, da forma mais simples

Feedback

Feedback contínuo do sistema, equipe e cliente

Coragem

Coragem para refatorar e fazer mudanças necessárias

Respeito

Respeito entre todos os membros da equipe

13 Práticas Principais

1. Jogo do Planejamento

Cliente define histórias de usuário

Equipe estima esforço necessário

2. Programação em Pares

Dois programadores, um computador

Melhora qualidade do código

3. Testes de Aceitação

Cliente define critérios de aceitação

Validação contínua

4. Design Simples

Sistema com design mais simples possível

5. Refatoração

Melhorar código sem alterar funcionalidade

6. Padronização do Código

Todos seguem mesmo padrão de codificação

7. Código Coletivo

Qualquer desenvolvedor pode alterar qualquer parte

8. Metáfora

Uso de metáforas para melhorar comunicação

9. Ritmo Sustentável

Sem horas extras excessivas

10. Integração Contínua

Sistema integrado várias vezes ao dia

11. Releases Curtos

Entregas frequentes de funcionalidades

12. Cliente Presente

Cliente disponível para tirar dúvidas

13. Desenvolvimento Orientado a Testes

Testes escritos antes do código

Destaque: Os testes são a base do desenvolvimento no XP!

Introdução ao FDD

Desenvolvimento Guiado por Funcionalidades

Origem

- Criado em 1997/1998
- Projeto para banco em Singapura
- Criadores: Peter Coad e Jeff De Luca

O que é?

Metodologia ágil que combina:

- Melhores práticas ágeis
- Engenharia de Software Orientada a Objetos
- Processos bem definidos

O que é uma Feature?

Definição

Feature = Requisito funcional de valor para o negócio

Características:

- Compreensível para clientes e usuários
- Implementável em até 2 semanas
- Fornece valor tangível
- Pode ser priorizado

Exemplo de Feature:

“Calcular o total de uma venda” “Validar login do usuário” “Gerar relatório mensal de vendas”

Vantagens do FDD

Para o Cliente/Gerente:

- Resultados significativos mais cedo
- Acompanhamento claro da evolução
- Priorização baseada em valor

Para os Desenvolvedores:

- Regras e técnicas fáceis de entender
- Resultados rápidos
- Processos bem definidos
- Foco em qualidade

Fase 1: Concepção e Planejamento

Duração: 1 a 2 semanas

Atividades:

- Pensar antes de fazer
- Modelagem do domínio
- Construção lista de features
- Planejamento por feature

Fase 2: Construção

Duração: Iterações de 2 semanas

Atividades:

- Fazer de forma iterativa
- Implementação das features
- Entregas frequentes

As 5 Atividades do FDD

1. **Desenvolver Modelo Abrangente**
 - Entender domínio do problema
2. **Construir Lista de Funcionalidades**
 - Identificar todas as features
3. **Planejar por Funcionalidade**
 - Priorizar e sequenciar features
4. **Detalhar por Funcionalidade**
 - Design detalhado de cada feature
5. **Construir por Funcionalidade**
 - Implementar, testar e integrar

O que torna FDD especial?

Iterativo e Incremental

Sistema evolui em pequenas partes

Blocos Pequenos

Features de 1-2 semanas

Ênfase na Qualidade

Testes e revisões constantes

Rastreabilidade

Acompanhamento preciso do progresso

Resultados Tangíveis

Entregas frequentes e funcionais

UP x XP x FDD

Aspecto	UP	XP	FDD
Foco	Processo estruturado	Práticas ágeis	Funcionalidades
Iterações	Por fases	1-3 semanas	2 semanas
Equipe	Variável	Até 12 pessoas	Pequena
Documentação	Completa	Mínima	Moderada
Mudanças	Controladas	Bem-vindas	Gerenciadas
Cliente	Envolvido	Presente	Prioriza

Recapitulando

Aprendemos:

UP: Processo estruturado em 4 fases (Concepção, Elaboração, Construção, Transição)

XP: 5 valores, 13 práticas, foco em qualidade e agilidade

FDD: Desenvolvimento guiado por features, resultados rápidos

Principais Conceitos:

- Metodologias podem ser combinadas
- Não existe “melhor metodologia”, apenas a mais adequada
- Foco na entrega de valor ao cliente

BONS ESTUDOS