UNIVERSIDADE PAULISTA - UNIP

**SEMINÁRIO SOBRE FEATURE DRIVEN DEVELOPMENT (FDD)**

R951952 - JEAN FLAVIO DE CAMPOS

H7703G3 - GABRIEL LIESEMBERG MASSARI

H775590 - GUSTAVO HENRIQUE DOS S MOREIRA

H7823F3 - RAMON GUIMARAES DE OLIVEIRA

R8975J5 - JOÃO PEDRO CAETANO

São Paulo - 2025

UNIVERSIDADE PAULISTA - UNIP

**SEMINÁRIO SOBRE FEATURE DRIVEN DEVELOPMENT (FDD)**

Trabalho apresentado à Universidade Paulista - UNIP, como requisito parcial para avaliação na disciplina de Engenharia de Software.  
  
Orientador: Prof. (Nome do orientador).

São Paulo - 2025

# SUMÁRIO

1. Introdução .................................................. 4

2. Contextualização .......................................... 5

3. Definição de FDD .......................................... 6

4. Princípios e Características ............................... 7

5. Processo do FDD ............................................ 8

6. Papéis no FDD .............................................. 9

7. Vantagens e Desvantagens .................................. 10

8. Exemplo Prático / Estudo de Caso .......................... 11

9. Comparação com Outras Metodologias Ágeis .................. 12

10. Conclusão ................................................ 13

11. Referências ............................................... 14

# 1. INTRODUÇÃO

Este trabalho tem como objetivo apresentar os conceitos fundamentais do Feature Driven Development (FDD), uma metodologia ágil orientada a funcionalidades. Serão explorados o contexto histórico, princípios, processos, papéis envolvidos, vantagens e desvantagens, bem como um exemplo prático de aplicação. O trabalho foi desenvolvido como parte das atividades acadêmicas da disciplina de Engenharia de Software da Universidade Paulista - UNIP.

# 2. CONTEXTUALIZAÇÃO

O desenvolvimento ágil é um conjunto de métodos e práticas voltados para a criação de software (e também aplicados em outros tipos de projetos) que priorizam a flexibilidade, a colaboração e a entrega contínua de valor ao cliente. Diferente dos métodos tradicionais, como o modelo cascata, que eram muito burocráticos e demorados, o ágil surgiu como uma resposta às dificuldades de adaptação a mudanças e ao tempo de entrega elevado.

A base do desenvolvimento ágil está no Manifesto Ágil (2001), que estabelece quatro valores principais: indivíduos e interações mais que processos e ferramentas; software em funcionamento mais que documentação abrangente; colaboração com o cliente mais que negociação de contratos; e responder a mudanças mais que seguir um plano.

O Feature Driven Development (FDD) faz parte do chamado 'guarda-chuva ágil', assim como o Scrum e o XP, seguindo os valores e princípios do Manifesto Ágil. No entanto, o FDD se diferencia por ser mais prescritivo e planejado, dando grande importância à modelagem inicial e ao design antes da implementação. Antes de começar a programar, a equipe constrói um modelo geral do sistema e uma lista detalhada de funcionalidades, que depois serão desenvolvidas em ciclos curtos de uma a duas semanas.

# 3. DEFINIÇÃO DE FDD

O Feature Driven Development (FDD), ou Desenvolvimento Guiado por Funcionalidades, é uma metodologia ágil focada na entrega frequente de pequenas funcionalidades que geram valor imediato ao cliente. A ideia é dividir grandes projetos em unidades menores e mais tangíveis, chamadas de 'features', facilitando a previsibilidade e o acompanhamento do progresso.

O FDD foi criado entre 1997 e 1999 por Jeff De Luca e Peter Coad. De Luca, consultor e arquiteto de software, liderava um grande projeto de reformulação da plataforma de empréstimos do United Overseas Bank, em Singapura. Dada a complexidade do projeto, ele convidou Peter Coad, especialista em modelagem orientada a objetos, para auxiliar na estruturação. Juntos, desenvolveram um processo disciplinado, mas ainda ágil e adaptável, baseado em funcionalidades pequenas, claras e entregáveis.

# 4. PRINCÍPIOS E CARACTERÍSTICAS

Esta seção aborda os princípios do FDD, como orientação a funcionalidades incrementais, modelagem orientada a objetos, qualidade do código e papéis bem definidos.

# 5. PROCESSO DO FDD

O processo do FDD é composto por cinco etapas: desenvolver um modelo geral, construir uma lista de funcionalidades, planejar por funcionalidade, detalhar por funcionalidade e construir por funcionalidade.

# 6. PAPÉIS NO FDD

Entre os principais papéis no FDD estão o Chief Architect, Chief Programmer, Class Owner, Domain Expert e Project Manager.

# 7. VANTAGENS E DESVANTAGENS

O FDD apresenta vantagens como entregas rápidas, previsibilidade e clareza no planejamento, mas também possui desvantagens, como menor flexibilidade para mudanças radicais e maior complexidade em equipes pequenas.

# 8. EXEMPLO PRÁTICO / ESTUDO DE CASO

Um exemplo real da aplicação do FDD ocorreu em 1997, quando o United Overseas Bank (UOB), de Singapura, decidiu reformular sua plataforma de empréstimos. Por ser um projeto de grande porte, o desenvolvimento foi dividido em funcionalidades independentes, o que permitiu que a plataforma continuasse operando durante a transição.

Como exemplo prático fictício, podemos considerar o desenvolvimento de um aplicativo de venda de roupas. Nesse caso, as áreas principais do sistema (usuários, produtos, carrinho de compras, pagamento e avaliações) seriam traduzidas em funcionalidades. Entre elas: cadastro de usuários, adição de produtos ao carrinho, finalização da compra com pagamento, avaliações e filtros de busca. Essas funcionalidades seriam planejadas em ciclos, atribuindo responsáveis a cada parte da equipe e garantindo entregas incrementais que já ofereçam valor desde os primeiros estágios.

# 9. COMPARAÇÃO COM OUTRAS METODOLOGIAS ÁGEIS

O FDD pode ser comparado a outras metodologias ágeis como o Scrum e o XP. Enquanto no Scrum a entrega é feita em sprints, que nem sempre resultam em funcionalidades isoladas utilizáveis, o FDD prioriza a entrega de features completas, permitindo que o cliente visualize o progresso de forma mais concreta. Em relação ao XP, o FDD é mais estruturado e enfatiza a modelagem inicial, sendo particularmente adequado a projetos grandes com muitas dependências e necessidade de previsibilidade.

# 10. CONCLUSÃO

Nesta seção, serão retomados os principais pontos apresentados ao longo do trabalho, destacando a relevância do FDD em determinados contextos de desenvolvimento.

# 11. REFERÊNCIAS

DE LUCA, J.; COAD, P. Java Modeling in Color with UML: Enterprise Components and Process. Prentice Hall, 1999.  
HIGHSMITH, J. Agile Software Development Ecosystems. Addison-Wesley, 2002.  
PRESSMAN, R. S. Engenharia de Software. 8ª Edição. McGraw Hill, 2016.  
Wikipedia. Feature Driven Development. Disponível em: https://pt.wikipedia.org/wiki/Feature\_Driven\_Development. Acesso em: 2025.