

Aula 2

Metodologias Ágeis

Prof. Manoel Flavio Leal

1

Conversa Inicial

2

Overview de métodos ágeis

- Métodos ágeis
 - Flexibilidade e adaptabilidade
 - Envolvimento dos clientes
 - Entrega frequente e software funcional

3

Estudaremos os métodos

- Extreme programming (XP)
- Feature Driven Development (FDD)
- Test Driven Development (TDD)
- Crystal
- Scrum

4

- Entenderemos suas origens
- Princípios, processos e benefícios
- Qual o método mais adequado para cada tipo de projeto

5

Extreme Programming (XP)

6

Conceitos

- Criado em 1996 por Kent Beck, Ward Cunningham e outros desenvolvedores
- Metodologia ágil de desenvolvimento de software
- Ênfase na simplicidade, comunicação e feedback rápido
- Visa produzir software de alta qualidade de forma eficiente

7

Valores fundamentais

- Comunicação: chave para resolver conflitos e identificar melhorias
- Simplicidade: busca soluções mais simples e organizadas
- Feedback: constante retorno dos clientes, colegas e usuários finais
- Coragem: tomada de decisões difíceis e disposição para mudanças

8

Princípios

- Feedback rápido: ajustes em tempo real com base no feedback do cliente
- Presumir simplicidade: soluções mais simples são mais fáceis de entender e manter
- Mudança incremental: desenvolvimento progressivo com adaptação contínua
- Trabalho de alta qualidade: testes, organização e documentação garantem qualidade

9

Princípios

- Pequenos passos: entrega de valor ao cliente em ciclos curtos
- Melhoria contínua: busca constante de aprimoramento do processo de desenvolvimento
- Diversidade: habilidades e experiências diversas para soluções inovadoras
- Reflexão constante: análise e ajuste após cada iteração do projeto

10

Princípios

- Comunicação: clareza e direção nas interações entre equipe e cliente
- Respeito: ambiente de trabalho agradável e produtivo com respeito às opiniões
- Coragem: tomar decisões ousadas e realizar mudanças necessárias
- Cliente presente: participação ativa e disponibilidade do cliente durante o projeto

11

Práticas do XP

- Jogo do planejamento: definição de funcionalidades em conjunto com o cliente
- Pequenos releases: entrega frequente de funcionalidades de valor
- Metáfora: uso de metáforas para facilitar o entendimento do software

12

Práticas do XP

- **Projeto simples:** construção de um software claro e objetivo
- **Testes:** criação de testes automatizados para garantir a qualidade

13

Práticas do XP

- **Programação em par:** pair programming para melhor colaboração e qualidade do código
- **Refatoração:** melhoria contínua do código
- **Integração contínua:** teste e validação frequentes do código
- **Ritmo sustentável:** estabelecimento de um ritmo de trabalho saudável
- **Cliente presente:** participação ativa do cliente no desenvolvimento

14

Práticas do XP

- **Programação em duplas rotativas:** rotação das duplas de programação para maior colaboração
- **Propriedade coletiva do código:** responsabilidade compartilhada pelo código
- **Padrões de codificação:** uso de padrões para facilitar a manutenção do código
- **Integração do usuário:** utilização de histórias de usuário para documentar as funcionalidades

15

Práticas do XP

- **Refatoração em larga escala:** melhoria do código em grande escala
- **Programa o que você precisa hoje:** desenvolvimento apenas do necessário para o momento
- **Uso de metodologias ágeis:** combinação com outras metodologias ágeis, como Scrum, Kanban e Lean

16

- **XP é uma metodologia ágil** focada em simplicidade, comunicação, feedback e coragem
- **Valores, princípios e práticas** orientam a equipe do projeto para entregar valor ao cliente
- **A equipe pode ajustar as práticas do XP** de acordo com suas necessidades
- **XP pode ser combinado com outras metodologias ágeis** para obter melhores resultados

17

Desenvolvimento dirigido por funcionalidades (FDD)

18

Conceitos

- **FDD (feature-driven development)** é uma abordagem ágil criada por Jeff De Luca e Peter Coad na década de 1990
- Ênfase na entrega rápida de funcionalidades de valor para o cliente
- Processo estruturado e orientado a processo

19

Etapas do FDD



Fonte: Leal, 2023

20

Etapas do FDD

- **Etapa 1: desenvolvimento do modelo geral**
 - Construção do modelo que apresenta a estrutura geral do sistema e identificação dos requisitos funcionais
- **Etapa 2: criar lista de funcionalidades**
 - Elaboração da lista de funcionalidades em conjunto com o cliente e sua priorização

21

Etapas do FDD

- **Etapa 3: planejar por funcionalidade**
 - Planejamento das tarefas, estimativa de tempo e recursos necessários para cada funcionalidade
- **Etapa 4: modelar por funcionalidade**
 - Construção do design detalhado de cada funcionalidade, incluindo especificação da arquitetura e diagramas

22

Etapas do FDD

- **Etapa 5: construir por funcionalidade**
 - Implementação e teste das funcionalidades em pequenos incrementos, repetindo até que todas sejam desenvolvidas e integradas

23

Características do FDD

- Foca nas funcionalidades
- Equipes multidisciplinares
- Processo iterativo e incremental



Diyajyoti /Shutterstock

24

Características do FDD

- Ênfase no planejamento
- Inspeções de código
- Ênfase na comunicação
- Entrega contínua

25

Vantagens do FDD

- Abordagem orientada a objetos
- Equipe multidisciplinar
- Entrega contínua de recursos
- Adaptabilidade

26

Desvantagens do FDD

- Falta de foco em documentação
- Dependência de habilidades técnicas
- Necessidade de planejamento detalhado
- Potencial para conflitos

27

- O (FDD) é uma abordagem ágil que se destaca pela entrega rápida e de valor para o cliente
- Suas etapas características, como o foco nas funcionalidades, equipes multidisciplinares e entrega contínua, contribuem para o desenvolvimento de software de alta qualidade
- É importante considerar as desvantagens do FDD, como a falta de foco em documentação e a necessidade de um planejamento detalhado, especialmente em projetos de grande escala

28

Desenvolvimento orientado a testes (TDD)

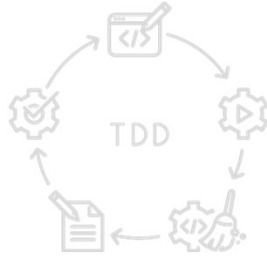
29

Conceitos

- TDD foi regulamentado e difundido nos anos 1990 por Kent Beck, como parte da metodologia Extreme Programming (XP)
- Surge como uma prática do desenvolvimento ágil para minimizar bugs e assegurar a qualidade do software
- Consiste em escrever testes antes mesmo de escrever o código em si

30

Como funciona o TDD?



Como funciona o TDD?

- Segue os seguintes passos
 - Escrever um teste
 - Escrever o código mínimo necessário para o teste passar
 - Executar o teste
 - Refatorar o código, sem alterar a funcionalidade
 - Repetir o processo para cada funcionalidade

31

32

Características do TDD

- Foco em testes
- Desenvolvimento iterativo
- Código mais seguro
- Refatoração constante
- Equipe mais produtiva

Vantagens do TDD

- Redução do número de defeitos
- Melhoria na qualidade do código
- Facilita a detecção de regressões
- Proporciona maior confiança no software

33

34

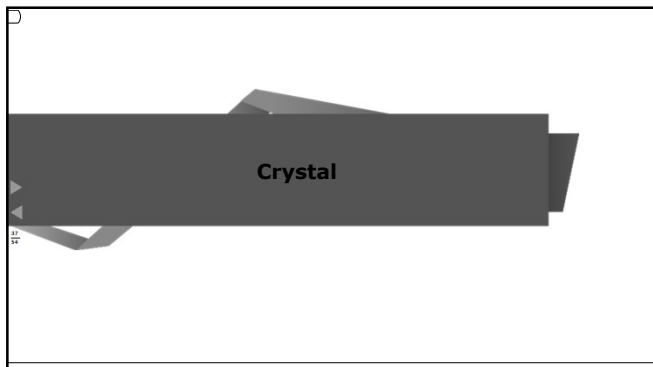
Desvantagens do TDD

- Possível aumento no tempo de desenvolvimento
- Possível aumento da complexidade do código
- Requer maior habilidade do desenvolvedor

- O TDD é uma prática que visa garantir a qualidade do software por meio da escrita de testes automatizados
- Possui vantagens como a redução de defeitos e melhoria na qualidade do código
- Apresenta desvantagens como o possível aumento no tempo de desenvolvimento
- É importante avaliar sua adequação às necessidades do projeto e à equipe de desenvolvimento

35

36



37

Conceitos

- Desenvolvida por Alistair Cockburn no final dos anos 1990
- Enfatiza a simplicidade, o trabalho em equipe, a comunicação e a capacidade de adaptação
- Proporciona um conjunto de diretrizes flexíveis e ajustáveis

38

Pilares da Crystal

- Comunicação
- Refinamento constante
- Entrega frequente
- Ambiente seguro
- Foco nas pessoas
- Acesso fácil
- Habilidade técnica

39

Família Crystal

- Cada método da família Crystal é identificado por uma cor e é recomendado para um determinado número de membros da equipe
- A família Crystal enfatiza a flexibilidade e a adaptação, permitindo ajustes de acordo com as características e requisitos de cada equipe

40

Tamanhos de equipe

- Crystal Clear: equipes de até 6 pessoas (com exceção de até 12 integrantes)
- Crystal Yellow: equipes de 10 a 20 participantes
- Crystal Orange: equipes de 20 a 50 pessoas
- Crystal Red: equipes de 50 a 100 membros

41

Crystal Clear

- Metodologia leve para equipes pequenas e colocalizadas
- Ênfase na comunicação cara a cara, colaboração, simplicidade e qualidade técnica
- Atividades executadas em incrementos de 1 a 3 meses

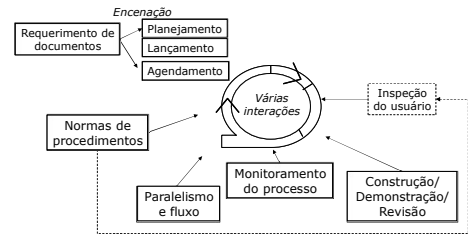
42

Crystal Orange

- Indicado para projetos de médio porte e mais de uma equipe multifuncional
- Ênfase na comunicação, colaboração, prototipagem rápida, testes frequentes e qualidade técnica

43

Ciclo de vida da Família Crystal



Fonte: Leal, 2023

44

Vantagens da Crystal

- Adaptação à equipe e flexibilidade
- Foco nas pessoas e colaboração
- Simplificação do processo

45

Desvantagens da Crystal

- Falta de estrutura para projetos complexos
- Menor documentação e compartilhamento de conhecimento
- Necessidade de expertise em desenvolvimento de software
- Controle de qualidade menos rigoroso

46

- Recapitulação dos principais pontos da Metodologia Crystal
- Destaque para a sua flexibilidade, adaptabilidade e foco nas pessoas
- Enfatizar a importância de selecionar a abordagem adequada para cada projeto e equipe

47

Scrum

48

Conceitos

- O Scrum é um framework ágil de gerenciamento de projetos
- Criado por Jeff Sutherland, John Scumniotales e Jeff McKenna no início dos anos 1990
- Inicialmente implantado em empresas de tecnologia nos Estados Unidos

49

Objetivos do Scrum

- Entregar valor ao cliente
- Adaptação às mudanças
- Maximizar a colaboração
- Melhoria contínua

50

Características do Scrum

- Time auto-organizado
- Ciclos de trabalho curtos (sprints)
- Backlog do produto
- Reuniões diárias (daily scrum)
- Sprint review
- Retrospectiva da sprint
- Papéis definidos
- Transparência

51

Vantagens do Scrum

- Maior flexibilidade
- Foco no valor para o cliente
- Transparência
- Colaboração
- Melhoria contínua

52

Desvantagens do Scrum

- Difícil de entender e implementar
- Requer muita disciplina e comprometimento
- Exige um alto nível de comunicação
- Não é adequado para todos os projetos
- Difícil de medir o progresso

53

- O Scrum é um framework ágil e produtivo para gerenciamento de projetos complexos
- Proporciona entrega de valor incremental e adaptativo aos clientes
- Foca na colaboração, autonomia e melhoria contínua da equipe
- Permite rápida adaptação às mudanças e necessidades do mercado
- Apesar dos desafios, o Scrum é amplamente adotado e eficaz em diversas áreas

54

