

Introdução ao Desenvolvimento Ágil

Prof. Me. Lucas Bruzzone

Aula 04

- **Rigidez:** Dificuldade para mudanças
- **Burocracia:** Excesso de documentação
- **Lentidão:** Processos demorados
- **Baixa adaptabilidade:** Não respondem bem a mudanças
- **Foco na documentação:** Não no software funcionando
- **Negociação contratual:** Mais importante que colaboração

Necessidade de uma nova abordagem

O Manifesto Ágil (2001)

Contexto

17 desenvolvedores se reuniram em Utah para discutir alternativas aos processos tradicionais

Resultado: Manifesto para Desenvolvimento Ágil de Software

Objetivo: Descobrir melhores formas de desenvolver software

Estamos descobrindo maneiras melhores de desenvolver software...

- **Indivíduos e interações** mais que processos e ferramentas
- **Software funcionando** mais que documentação abrangente
- **Colaboração com cliente** mais que negociação contratual
- **Responder a mudanças** mais que seguir um plano

Valorizamos os itens à direita, mas valorizamos mais os itens à esquerda

Os 12 Princípios Ágeis - Parte 1

- 1 Nossa maior prioridade é **satisfazer o cliente** através da entrega contínua e adiantada de software com valor agregado
- 2 Mudanças nos requisitos são **bem-vindas**, mesmo tardiamente no desenvolvimento
- 3 **Entregar frequentemente** software funcionando, de poucas semanas a poucos meses
- 4 Pessoas de negócio e desenvolvedores devem **trabalhar diariamente** em conjunto
- 5 Construa projetos em torno de **indivíduos motivados**. Dê a eles o ambiente e o suporte necessário

Os 12 Princípios Ágeis - Parte 2

- ⑥ O método mais eficiente e eficaz de transmitir informações é através de **conversa face a face**
- ⑦ **Software funcionando** é a medida primária de progresso
- ⑧ Os processos ágeis promovem **desenvolvimento sustentável**
- ⑨ Contínua atenção à **excelência técnica** e bom design
- ⑩ **Simplicidade** - a arte de maximizar a quantidade de trabalho não realizado
- ⑪ As melhores arquiteturas emergem de **equipes auto-organizáveis**
- ⑫ A equipe **reflete regularmente** sobre como se tornar mais eficaz

Desenvolvimento

- Iterativo e incremental
- Entregas frequentes
- Adaptativo
- Time-boxing
- Feedback contínuo

Equipe

- Auto-organizável
- Multifuncional
- Colaborativa
- Comunicação face a face
- Cliente integrado

- **Planning Poker:** Estimativa colaborativa
- **Stand-up meetings:** Reuniões diárias rápidas
- **Retrospectivas:** Reflexão e melhoria contínua
- **User Stories:** Requisitos na perspectiva do usuário
- **Burndown charts:** Acompanhamento visual do progresso
- **Pair programming:** Programação em dupla
- **Refactoring:** Melhoria contínua do código
- **Test-driven development:** Testes antes do código

Benefícios das Metodologias Ágeis

- **Entrega de valor cedo:** Software funcionando rapidamente
- **Flexibilidade:** Adapta-se bem a mudanças
- **Qualidade:** Testes e refatoração contínuos
- **Satisfação do cliente:** Participação ativa no desenvolvimento
- **Motivação da equipe:** Autonomia e auto-organização
- **Redução de riscos:** Feedback frequente e entregas curtas
- **Transparência:** Progresso visível

Desafios das Metodologias Ágeis

- **Mudança cultural:** Requer nova mentalidade
- **Disciplina:** Precisa de equipe madura
- **Documentação:** Pode ser insuficiente para alguns contextos
- **Escalabilidade:** Difícil em projetos muito grandes
- **Contratos:** Conflito com modelos contratuais tradicionais
- **Métricas:** Medição de progresso diferente
- **Resistência:** Organizações resistentes a mudanças

- **Scrum:** Framework para gestão de projetos ágeis
- **Extreme Programming (XP):** Foca em práticas de desenvolvimento
- **Kanban:** Visualização e gestão do fluxo de trabalho
- **Lean Software Development:** Baseado nos princípios Lean
- **Crystal:** Família de metodologias adaptáveis
- **Feature Driven Development (FDD):** Orientado a funcionalidades

Cada metodologia tem características específicas

Quando Usar Metodologias Ágeis?

Adequado quando:

- Requisitos podem mudar
- Equipe pequena a média (2-12 pessoas)
- Cliente disponível para colaborar
- Tecnologia conhecida pela equipe
- Organização flexível
- Projeto inovador

Cuidado quando:

- Requisitos muito estáveis
- Equipes muito grandes
- Regulamentações rígidas
- Contratos de escopo fechado

Scrum: Práticas e Artefatos

Estudaremos detalhadamente o framework Scrum, seus papéis, eventos e artefatos