

Métodos Ágeis

Prof.^a Ma. Jessica Oliveira

Aula 02 – 28/02/2025

Manifesto Ágil e Princípios Fundamentais.

História e criação do Manifesto Ágil

- Nos anos 1990, o desenvolvimento de *software* enfrentava desafios significativos devido à rigidez dos modelos tradicionais. **Projetos demoravam anos para serem concluídos, e qualquer mudança nos requisitos resultava em altos custos e atrasos.**
- Como resposta a esses problemas, um grupo de **17 desenvolvedores e especialistas em *software*** se reuniu em fevereiro de 2001 no *resort Snowbird*, em Utah, nos Estados Unidos, para discutir novas abordagens.

História e criação do Manifesto Ágil

- Os signatários do Manifesto Ágil incluíam figuras como Kent Beck, Robert C. Martin, Martin Fowler e Jeff Sutherland, criadores de metodologias como *Extreme Programming* (XP), *Scrum* e *Crystal*.
- O principal objetivo do encontro era encontrar uma **alternativa aos processos pesados de desenvolvimento** e criar um conjunto de princípios que priorizassem a entrega de *software* de forma **eficiente e flexível**.
- O resultado dessa reunião foi o **Manifesto Ágil**, um documento conciso que define os **valores e princípios fundamentais** do desenvolvimento ágil.

Valores do Manifesto Ágil

Indivíduos e interações mais que processos e ferramentas.

- Foco na comunicação e colaboração entre os membros da equipe.
- Processos e ferramentas são úteis, mas não devem substituir a interação humana.
- **Exemplo:** um time ágil realiza reuniões diárias para alinhar prioridades ao invés de depender exclusivamente de *e-mails* ou sistemas formais de documentação.

Software funcionando mais que documentação abrangente.

- A prioridade é entregar um produto funcional o mais rápido possível.
- Documentação ainda é importante, mas não deve retardar o progresso do desenvolvimento.
- Exemplo: ao invés de gastar meses escrevendo um documento de requisitos detalhado, uma equipe ágil pode desenvolver uma versão mínima do *software* (MVP) e aprimorá-lo com base no *feedback* do usuário.

Colaboração com o cliente mais que negociação de contratos.

- O cliente deve estar envolvido durante todo o desenvolvimento.
- Requisitos podem mudar conforme as necessidades do negócio evoluem.
- **Exemplo:** no Scrum, o *Product Owner* representa o cliente e participa ativamente do desenvolvimento, priorizando funcionalidades conforme o feedback do mercado.

Responder a mudanças mais que seguir um plano.

- Planos são úteis, mas devem ser flexíveis para acomodar mudanças.
- A capacidade de adaptação é um diferencial competitivo.
- **Exemplo:** se um aplicativo de *e-commerce* precisa de uma funcionalidade emergencial devido a uma demanda do mercado, a equipe ágil pode priorizá-la sem comprometer o andamento do projeto.

Princípios do Manifesto Ágil

Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de *software* com valor.

- Em métodos ágeis, a satisfação do cliente é primordial. O *software* deve ser entregue de forma incremental e com funcionalidades úteis desde as primeiras versões.
- **Exemplo:** uma equipe desenvolvendo um aplicativo de compras online pode lançar inicialmente uma versão simples com catálogo de produtos, adicionando funcionalidades como carrinho e pagamento seguro nas iterações seguintes.

Aceitamos mudanças de requisitos, mesmo no final do desenvolvimento. Processos ágeis aproveitam mudanças para vantagem competitiva do cliente.

- Em projetos tradicionais, mudanças nos requisitos geralmente são vistas como problemas. No ágil, são bem-vindas e esperadas.
- **Exemplo:** se um *e-commerce* precisa integrar um novo método de pagamento popular, a equipe ágil pode replanejar a *sprint* para incluir essa mudança.

Entregamos *software* funcional frequentemente, de poucas semanas a poucos meses, com preferência a prazos mais curtos.

- A entrega contínua permite que os clientes utilizem e testem partes do *software* antes da versão final.
- **Exemplo:** em vez de lançar um sistema completo após um ano de desenvolvimento, uma empresa pode disponibilizar módulos individuais a cada três semanas.

Pessoas de negócio e desenvolvedores devem trabalhar juntas diariamente durante todo o projeto.

- O ágil reduz a distância entre os *stakeholders* e a equipe técnica, garantindo que as expectativas sejam alinhadas em tempo real.
- **Exemplo:** em equipes Scrum, o *Product Owner* (PO) representa o cliente e trabalha lado a lado com os desenvolvedores.

Construa projetos ao redor de indivíduos motivados. Dê a eles o ambiente e o suporte de que precisam, e confie que farão seu trabalho.

- A autonomia e o apoio adequado aumentam o engajamento e a produtividade das equipes.
- **Exemplo:** um time ágil pode definir suas próprias estratégias para organizar tarefas, em vez de receber ordens detalhadas de um gerente.

O método mais eficiente e eficaz de transmitir informações para e dentro de uma equipe de desenvolvimento é a conversa cara a cara.

- Embora ferramentas como *e-mails* e mensagens sejam úteis, a comunicação verbal direta reduz ambiguidades.
- **Exemplo:** reuniões diárias (*daily meetings*) garantem que todos estejam alinhados sobre prioridades e obstáculos.

Software funcional é a principal medida de progresso.

- Documentação e relatórios são importantes, mas o que realmente importa é um *software* funcionando.
- **Exemplo:** um *backlog* cheio de tarefas concluídas não tem valor se o *software* ainda não puder ser usado pelos clientes.

Processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

- O ritmo do projeto deve ser sustentável, evitando sobrecarga e esgotamento da equipe.
- **Exemplo:** no Scrum, a velocidade média da equipe (*velocity*) é monitorada para evitar demandas excessivas.

Atenção contínua à excelência técnica e bom *design* aumenta a agilidade.

- Código bem escrito e arquiteturas bem planejadas tornam o *software* mais flexível e fácil de manter.
- **Exemplo:** métodos como *Test-Driven Development* (TDD) ajudam a garantir que cada parte do *software* seja funcional antes mesmo de ser integrada ao sistema.

Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.

- No ágil, busca-se evitar desperdício, desenvolvendo apenas o que realmente agrega valor.
- **Exemplo:** em vez de criar uma funcionalidade extremamente complexa que pode nunca ser usada, a equipe prioriza as que geram impacto imediato.

As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

- No ágil, as equipes têm autonomia para definir a melhor forma de trabalhar, sem depender exclusivamente de gerentes.
- **Exemplo:** o time pode decidir se usará Scrum, XP ou outra abordagem com base no contexto do projeto.

Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e ajusta seu comportamento de acordo.

- A melhoria contínua é um pilar do ágil. Retrospectivas são realizadas para identificar o que pode ser aprimorado.
- **Exemplo:** se uma equipe percebe que há atrasos constantes na entrega de tarefas, pode decidir mudar a forma como organiza o *backlog*.

Impactos do Desenvolvimento Ágil na Indústria de *Software*.

Benefícios.

- **Redução do tempo de entrega:** ciclos curtos de desenvolvimento aceleram a disponibilização de novas funcionalidades.
- **Maior alinhamento com o cliente:** reuniões frequentes garantem que o produto atenda às necessidades do usuário final.
- **Menos desperdício:** o foco está em desenvolver apenas o que agrega valor.

Desafios.

- Empresas tradicionais podem encontrar resistência à mudança, pois o ágil exige uma cultura organizacional flexível.
- Equipes não acostumadas com metodologias ágeis podem ter dificuldade em se auto-organizar e priorizar tarefas corretamente.
- A falta de uma liderança bem estruturada pode resultar em falta de direcionamento para o time.

Relação com as competências da Engenharia de *Software*.

Habilidades técnicas exigidas no ágil.

- **Programação e refatoração contínua** (para adaptar o código às novas demandas).
- **Testes automatizados** (para garantir qualidade e reduzir retrabalho).
- **Integração e entrega contínuas** (CI/CD) (para manter o *software* atualizado constantemente).

Soft skills essenciais.

- **Comunicação e trabalho em equipe** (para interagir com clientes e stakeholders).
- **Capacidade de adaptação e resolução de problemas** (para responder rapidamente a mudanças).
- **Autogerenciamento e proatividade** (para contribuir com decisões técnicas sem precisar de ordens diretas).

O ágil como diferencial competitivo.

- Empresas valorizam profissionais que sabem trabalhar em metodologias ágeis, pois isso melhora a produtividade e reduz riscos.
- Certificações como *Certified Scrum Master* (CSM) e *Professional Scrum Developer* (PSD) aumentam a empregabilidade do profissional.

Vamos para a prática?

Fábrica Ágil de Aviões (Parte 1).

Fundando a Fábrica.

- Dividam-se em equipes para trabalharem juntos durante o semestre, não sendo necessário atribuir funções a ninguém, por enquanto;
- Deem um nome para a fábrica;
- Criem o Manifesto Ágil da fábrica baseado nos princípios do Manifesto Ágil original.

Para o pós-aula...

- Escrever um texto refletindo sobre como os quatro valores do Manifesto Ágil poderiam ser aplicados em um projeto real.
- A atividade estará detalhada no AVA.



Dúvidas?

jessica.oliveira@p.ucb.br