

ENGENHARIA DE SOFTWARE

Desenvolvimento Ágil

Kele Teixeira Belloze
kelebelloze@gmail.com

PROBLEMAS RECORRENTES NO DESENVOLVIMENTO DE SOFTWARE

- Processo de desenvolvimento de software é caótico
- Imprevisibilidade e incertezas fazem parte desse processo
- Requisitos de usuário, pressões de tempo, competição, qualidade e recursos (base inicial do projeto) podem sofrer alterações
- Problema de modelos lineares (cascata, ...): assumir o processo como não sujeito a mudanças e incertezas



AGILIDADE (1/2)

- O que é agilidade?
 - “Habilidade de criar e responder à mudanças de maneira a aproveitar as mudanças no ambiente”
- “Linha de pensamento” revolucionária
 - Precisamos parar de tentar evitar mudanças



AGILIDADE (2/2)

○ Metodologias ágeis

- Coleção de **práticas ágeis** organizadas para modelagem e desenvolvimento de software
- A especificação, a modelagem e a implementação são **simultâneos**. (Não são uma sequência de etapas distintas executadas em série).
- Procuram **minimizar riscos** desenvolvendo software em **pequenos espaços de tempo** (iterações)
- O sistema é desenvolvido como uma **série de versões com stakeholders envolvidos na validação** de versões.

- **“Filosofia” onde muitas “metodologias” se encaixam**
- **Definem um conjunto de atitudes e não processo prescritivo**



MOTIVAÇÃO (1/2)

- A insatisfação causada pelas sobrecargas (no planejamento, projeto e documentação) nos métodos orientados a planos das décadas de 1980 e 1990 levaram a criação dos métodos ágeis.

Os métodos ágeis:

- Focam mais no código que na modelagem.
- São baseados numa abordagem iterativa de desenvolvimento de software.
- Pretendem entregar software funcional de forma rápida e evoluir rapidamente para atender aos requisitos



MOTIVAÇÃO (2/2)

- A meta dos métodos ágeis é:
 - Reduzir sobrecargas no processo de software (limitando a documentação, por exemplo) e,
 - Possibilitar uma resposta rápida a mudanças de requisitos sem gerar um retrabalho excessivo.

Manifesto ágil:

- Representa a filosofia por trás do método ágil.



REAÇÃO ÀS METODOLOGIAS TRADICIONAIS

○ “Manifesto ágil” (2001)

- Princípios

- **Indivíduos e iterações** são mais importantes que processos e ferramentas
- **Software funcionando** é mais importante que documentação completa
- **Colaboração com o cliente** é mais importante que negociação com contratos
- **Adaptação às mudanças** é mais importante que seguir um plano



PRINCÍPIOS DOS MÉTODOS ÁGEIS

Princípio	Descrição
Envolvimento do cliente	Clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos e o cliente especifica os requisitos a serem incluídos em cada incremento.
Pessoas ao invés de processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritos.
Atender mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças
Manter simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

APLICAÇÃO

- Desenvolvimento de produtos para venda de pequeno ou médio porte.
- Desenvolvimento de sistemas customizados em uma organização onde há um compromisso claro do cliente de se envolver no processo de desenvolvimento e onde não existe um grande número de regras externas e regulamentos que afetem o software.
- Por causa do foco em equipes pequenas e fortemente integradas existem problemas para escalar métodos ágeis em sistemas de grande porte.



PROBLEMAS

- Pode ser difícil manter o interesse do cliente que está envolvido no sistema.
- Membros de equipes podem não se adequar ao intenso envolvimento que caracteriza os métodos ágeis.
- A priorização de mudanças pode ser difícil quando se tem vários *stakeholders*.
- Manter a simplicidade requer trabalho extra.
- Contratos podem ser um problema como em outras abordagens de desenvolvimento iterativo.

