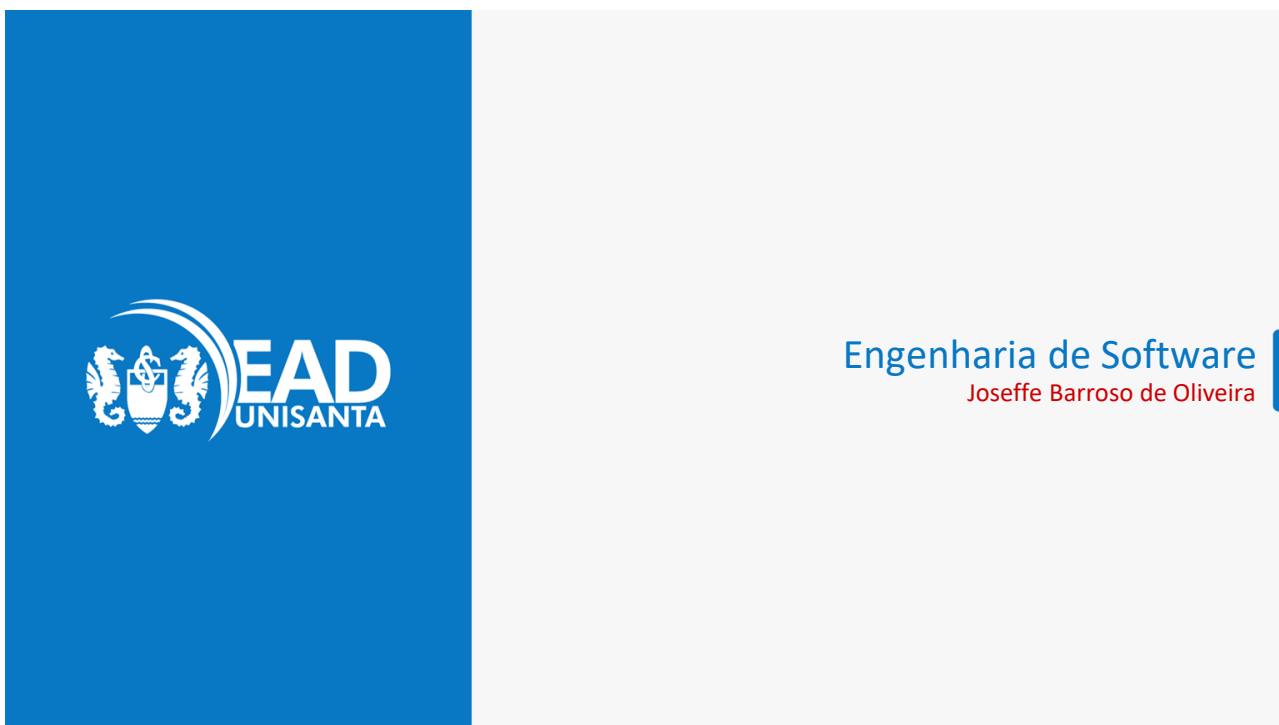




1



2

Introdução

Em 2001, um grupo composto por desenvolvedores de software, autores e consultores de renome assinou o “Manifesto para o desenvolvimento ágil de software” (“Manifesto for Agile software Development”), no qual defendiam **“indivíduos e interações acima de processos e ferramentas, software operacional acima de documentação completa, colaboração dos clientes acima de negociação contratual e respostas a mudanças acima de seguir um plano”**.

As ideias fundamentais que norteiam o desenvolvimento ágil levaram ao desenvolvimento dos métodos ágeis, projetados para sanar fraquezas, supostas e reais, da engenharia de software convencional.

O que é agilidade?

Afinal, o que é agilidade no contexto da engenharia de software? Ivar Jacobson argumenta que a **difusão da mudança** é o principal condutor para a agilidade. Os engenheiros de software **devem ser rápidos** caso queiram **assimilar as rápidas mudanças** que Jacobson descreve.

Entretanto, agilidade é mais do que uma resposta à mudança. Ela abrange também a filosofia proposta no manifesto. Ela incentiva a **estruturação e as atitudes em equipe que tornam a comunicação mais fácil** (entre membros da equipe, entre o pessoal ligado à tecnologia e o pessoal da área comercial, entre os engenheiros de software e seus gerentes).



5

O que é agilidade?

Enfatiza a entrega rápida do software operacional e diminui a importância dos artefatos intermediários (nem sempre um bom negócio); aceita o cliente como parte da equipe de desenvolvimento e trabalha para **eliminar a atitude de “nós e eles”** que continua a impregnar muitos projetos de software; reconhece que o planejamento em um mundo incerto tem seus limites, e que o plano (roteiro) de projeto deve ser flexível.

A agilidade pode ser aplicada a qualquer processo de software. No entanto, para alcançá-la, é essencial que o processo seja projetado de modo que a equipe possa adaptar e alinhar (racionalizar) tarefas.



6

Agilidade e o custo das mudanças

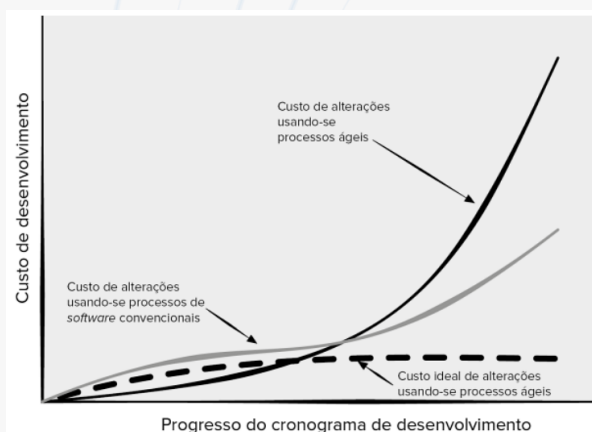
O senso comum no desenvolvimento de software (baseado em décadas de experiência) é que **os custos de mudanças aumentam** de forma não linear conforme o projeto avança. É relativamente fácil acomodar uma mudança quando a equipe de software está reunindo requisitos (no início de um projeto). Os custos desse trabalho são mínimos, e o tempo demandado não afetará negativamente o resultado do projeto. Mas, se adiantarmos alguns meses, o que aconteceria? A equipe está em meio aos testes de validação (que ocorrem relativamente no final do projeto), e um importante envolvido está solicitando uma mudança funcional grande. Os custos crescem rapidamente, e o tempo e os esforços necessários para assegurar que a mudança seja feita sem efeitos colaterais inesperados não serão insignificantes.



7

Agilidade e o custo das mudanças

Os proponentes da agilidade argumentam que um processo ágil bem elaborado “achata” o custo da curva de mudança, permitindo que uma equipe de software assimile as alterações, realizadas posteriormente em um projeto de software, sem um impacto significativo nos custos ou no tempo.



8

O que é processo ágil?

Qualquer processo ágil de software é caracterizado de uma forma que trate de uma série de preceitos-chave acerca da maioria dos projetos de software:

1. É difícil prever quais requisitos de software vão persistir e quais sofrerão alterações.
2. Para muitos tipos de software, o projeto e a construção são intercalados.
3. Análise, projeto, construção (desenvolvimento) e testes não são tão previsíveis (do ponto de vista de planejamento) quanto gostaríamos que fossem.

Dados esses três preceitos, surge uma importante questão: **como criar um processo capaz de administrar a imprevisibilidade?**



O que é processo ágil?

A resposta, conforme já observado, **está na adaptabilidade do processo** (alterar rapidamente o projeto e as condições técnicas). Portanto, um processo ágil deve ser adaptável.

Adaptação contínua sem progressos, entretanto, de pouco adianta. Um processo ágil de software deve adaptar de modo incremental. Para conseguir uma adaptação incremental, a equipe ágil precisa de feedback do cliente (de modo que as adaptações apropriadas possam ser feitas).

Essa abordagem iterativa capacita o cliente a avaliar o incremento de software regularmente, fornecer o feedback necessário para a equipe de software e influenciar as adaptações feitas no processo para incluir o feedback adequadamente.



Princípios de agilidade

A Agile Alliance estabelece 12 princípios para alcançar a agilidade. São eles:

