

Residência em TI Gerência de Configuração e Teste de Software

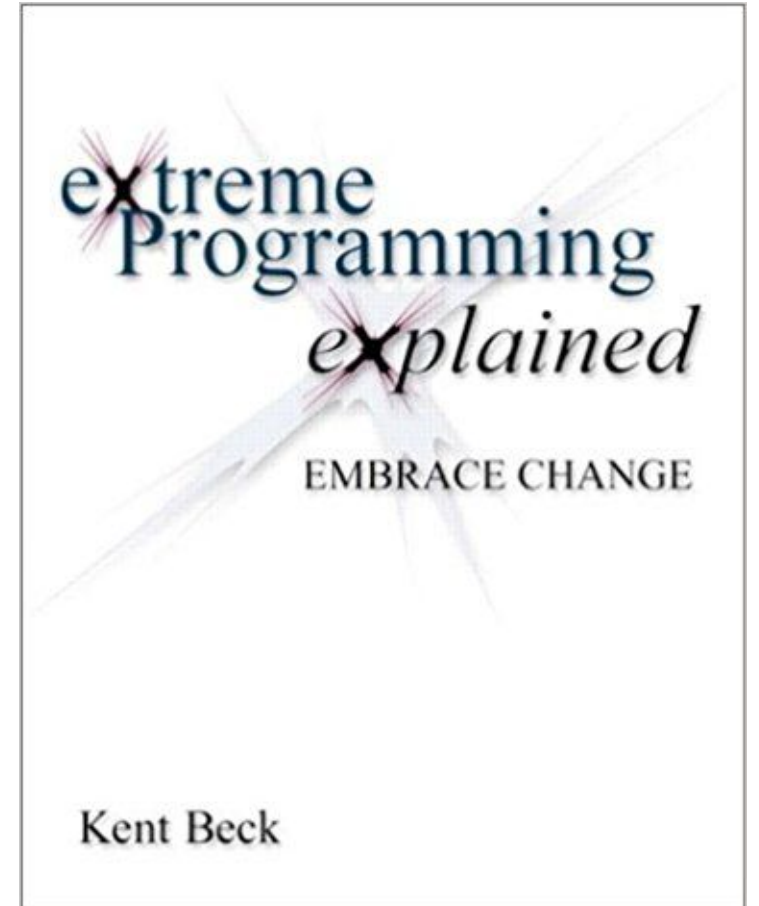
Prof. Eiji Adachi

Objetivos

- Apresentar a técnica Test-Driven Development

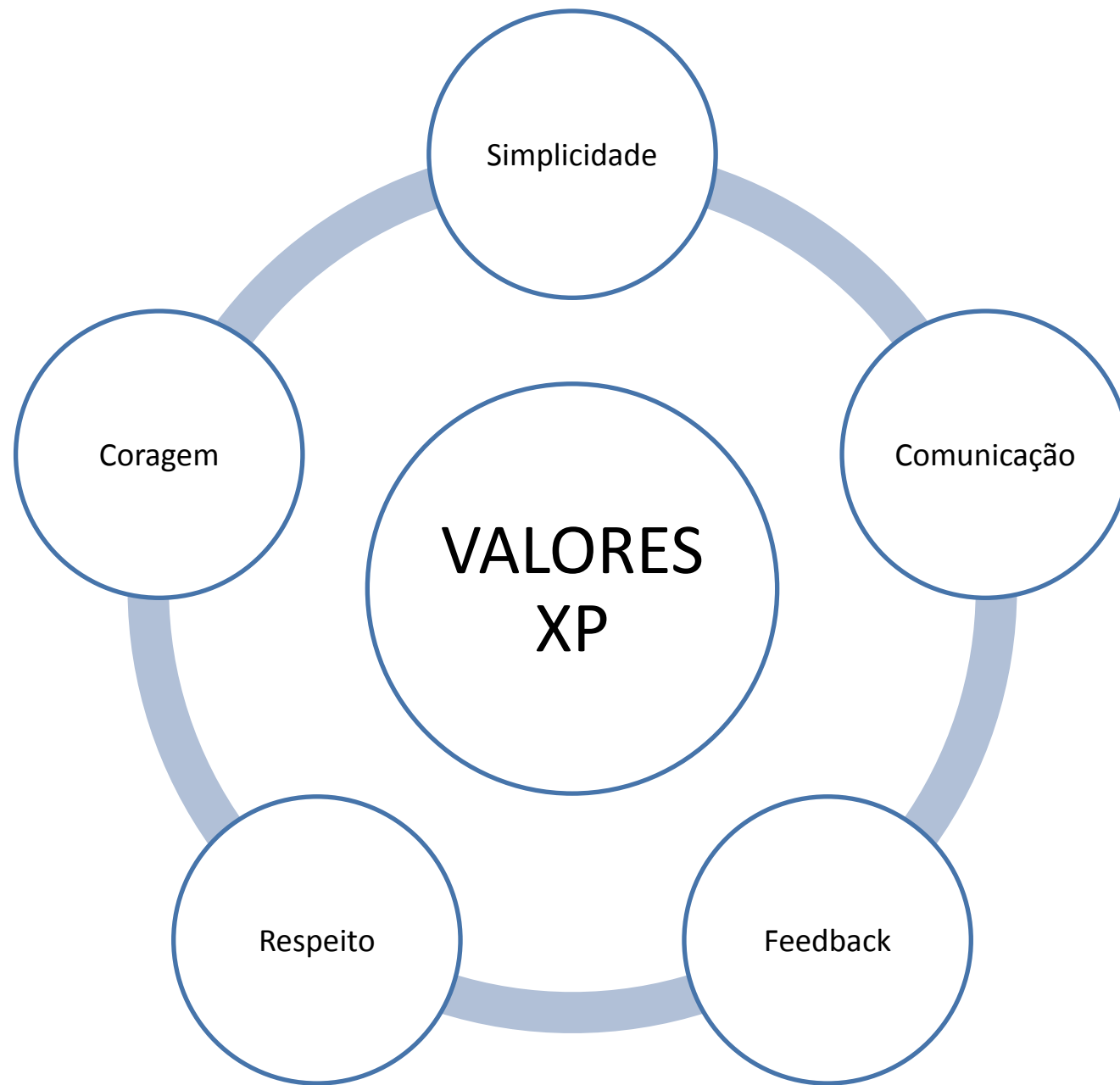
Contextualização

- Método eXtreme Programming (XP) foi criado por Kent Beck nos anos 1990
- Livro publicado em 1999



Ideia central do XP

- Se uma prática é eficaz, vamos empregá-la ao extremo



Práticas

- Jogo do planejamento
- Entregas pequenas
- Metáfora
- Design simples
- **Desenvolvimento dirigido a testes**
- **Refatoração**
- Programação em pares
- Propriedade coletiva
- Integração contínua
- Semana de 40 horas
- Cliente no local
- Padrões de código
- Espaço de trabalho aberto

Test-Driven Development (TDD)

- Prática de desenvolvimento de software dirigida a testes
 - Testes unitários são criados antes das funcionalidades serem implementadas
 - Testes unitários são executados sistematicamente
- Prática popularizada por métodos ágeis, como o Extreme Programming (XP)

Como funciona?

- Antes de implementar uma funcionalidade:
 - Especifique o comportamento esperado para esta funcionalidade
 - Escreva testes unitários que exercitem estes comportamentos especificados
- Durante a implementação da funcionalidade:
 - Execute os testes até que todos passem sem falhas
 - Refatore o código para melhorar sua estrutura

Refatoração

- Reestruturar a “parte interna” do sistema, sem alterar seu comportamento
- Reestruturar removendo duplicação e complexidade, visando melhorar a comunicação e a flexibilidade do código

Ciclo TDD

- Selecione uma funcionalidade:
 - Especifique o comportamento esperado desta funcionalidade
 - Crie casos de testes
 - Compile
 - Corrija os erros de compilação
 - Execute os casos de teste e os veja falhar (RED)
 - Implemente a funcionalidade
 - Execute os casos de teste e os veja passar (GREEN)
 - Refatore

Por que empregar?

- Garante que toda funcionalidade será testada
- Ajuda a estruturar o software de modo mais fácil de testar
- Aumenta a confiança na hora de realizar mudanças no código
 - Apoia tarefas de refatoração

Exercício

Use o TDD para implementar a funcionalidade de consolidar parcialmente uma turma. As regras para definir o status parcial são:

O regulamento dos cursos de graduação da UFRN define uma série de artigos que especificam os critérios de aprovação para disciplinas regulares presenciais. Dentre estes, os seguintes se destacam:

Art. 94. A aprovação em um componente curricular está condicionada à obtenção do rendimento acadêmico mínimo exigido na avaliação da aprendizagem e, para os componentes curriculares presenciais, à frequência mínima exigida na avaliação da assiduidade.

Art. 98. O rendimento acadêmico nas disciplinas e módulos deve ser expresso em valores numéricos de 0 (zero) a 10 (dez), variando até a primeira casa decimal, após o arredondamento da segunda casa decimal.

Art. 99. Com o fim de sistematizar as atividades a serem desenvolvidas na disciplina ou módulo, o período letivo é dividido em 3 (três) unidades.

Art. 104. Em cada componente curricular, a média parcial é calculada pela média aritmética dos rendimentos escolares obtidos em cada unidade.

Art. 105. É considerado aprovado, quanto à avaliação de aprendizagem, o estudante que satisfaz um dos seguintes critérios:

I – tem média parcial igual ou superior a 7,0 (sete); ou

II – tem média parcial igual ou superior a 5,0 (cinco), com rendimento acadêmico igual ou superior a 3,0 (três) em todas as unidades.

Art. 106. O estudante que não atinge os critérios de aprovação definidos no artigo 105 tem direito à realização de uma avaliação de reposição se todas as seguintes condições forem atendidas:

I – O critério de aprovação por assiduidade é satisfeito; e

II – O estudante tem média parcial igual ou superior a 3,0 (três).

Art. 113. Para ser aprovado em uma disciplina ou módulo presencial, o estudante deve comparecer a aulas que totalizem 75% (setenta e cinco por cento) ou mais da carga horária do componente curricular ou a 75% (setenta e cinco por cento) ou mais do total de aulas ministradas, o que for menor.

Residência em TI Gerência de Configuração e Teste de Software

Prof. Eiji Adachi