

# ENGENHARIA DE SOFTWARE III

## 1 TDD - Desenvolvimento Guiado por Testes

“O desenvolvimento dirigido a testes (TDD - Test-Driven Development) é uma abordagem para o desenvolvimento de softwares em que se intercalam testes e desenvolvimento de código” (SOMMERVILLE, 2011, p. 155). Para Kent (2010, p. 223), TDD é “[...] programação com teste primeiro”.

De modo mais detalhado, em uma abordagem mais atual, desenvolvimento de software pode ser compreendido como “[...] uma junção complexa de análise, projeto lógico, projeto físico, implementação, teste, revisão, integração e implantação” (KENT, 2010, p. 223). Diante disso, “[...] a antiga forma de pensar em desenvolvimento de software como fases está enfraquecida devido ao feedback entre as decisões ser difícil se elas estão separadas no tempo” (KENT, 2010, p. 223).

O termo guiado faz referência aos elementos que são utilizados para a base da elaboração do código, que podem ser especificações, especulações, ou no caso do TDD, os testes.

Quanto aos testes no TDD, eles são “[...] automatizados, materializados, concretos”, ou seja, basta um comando e “[...] eles executam. Uma das ironias de TDD é que não é uma técnica de teste. É uma técnica de análise, uma técnica de projeto, realmente uma técnica para estruturar todas as atividades de desenvolvimento” (KENT, 2010, p. 224).

“Um argumento forte a favor do desenvolvimento dirigido a testes é que ele ajuda os programadores a clarear suas ideias sobre o que um segmento de código supostamente deve fazer” (SOMMERVILLE, 2011, p. 155). Diante disso, “para escrever um teste”, o desenvolvedor “[...] precisa entender a que ele se destina”, e “[...] esse entendimento faz que seja mais fácil escrever o código necessário” (SOMMERVILLE, 2011, p. 155-156).

Conforme Kent (2010, p. 214), todo desenvolvedor “[...] deve testar condicionais, laços, operações e polimorfismo”.

Por fim, “[...] quanto mais cedo se encontrar e corrigir um defeito, mais barato é” (KENT, 2010, p. 222).

## 2 Refatoração

A refatoração visa uma otimização do código. Todavia, “uma refatoração não pode mudar a semântica do programa sob qualquer circunstância” (KENT, 2010, p. 201)

“A princípio,” em um processo de refatoração, deve-se “[...] estar preparado para dar muitos passos pequenos”, além de que deve ser realizado de modo minucioso, pois uma refatoração “está propensa a erros” (KENT, 2010, p. 212).

## 3 Depuração

“A depuração” é um processo que possibilita a “auditoria de código” (DOWD; MCDONALD; SCHUH, 2006, p. 153).

“O nível de sofisticação nos depuradores varia muito, assim como seus conjuntos de recursos” (DOWD; MCDONALD; SCHUH, 2006, p. 166).

## 4 Implantação de software

A Implantação de software “[...] é o processo de tornar o sistema disponível para os usuários, transferir dados dos sistemas existentes e estabelecer comunicações com outros sistemas no ambiente” (SOMMERVILLE, 2011, p. 195). Nesta etapa, “os usuários começam a usar o sistema para apoiar seu trabalho” (SOMMERVILLE, 2011, p. 195). Portanto, essa fase envolve a “instalação de software”, bem como o “[...] middleware e sua correta configuração para que as vulnerabilidades [...] sejam evitadas” (SOMMERVILLE, 2011, p. 256).

\* Apenas para fins educacionais.

## BIBLIOGRAFIA

DOWD, M. MCDONALD, J. SCHUH, J. ***The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities***. 1. ed. Boston: Addison-Wesley Professional, 2006.

KENT, B. **TDD: Desenvolvimento guiado por testes**. Porto Alegre: Bookman, 2010.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.