



Universidade de Brasília - UnB

Instituto de Exatas
Departamento de Ciência da Computação

Rodrigo Chaves - 13/0132624
Gabriel Mesquita - 13/0024242

Test-drive Development

Brasília - DF
2016

Gabriel Mesquita ...
Rodrigo de Araujo Chaves

Test-driven Development

Dissertação sobre por que Test-driven development
é pratica que melhorar a qualidade
final do software apresentanda à disciplina
de Engenharia de Software da Universidade de Brasília.

Brasília - DF
2016

1 Introdução

2 O que é Test-driven Development

Test-driven development é uma prática de desenvolvimento de software que tem sido usada esporadicamente por décadas. Com essa prática, um engenheiro de software passa por ciclos entre escrever um teste de unidade que falha e escrevendo a implementação do software para passar nesses testes. Test-driven development tem recentemente resurgido como uma prática crítica possibilitando metodologias de desenvolvimento ágil de software.

Quando discutimos sobre TDD, é considerado um conjunto de tarefas requeridas que podem ser implementadas em poucos dias ou menos. Na imagem 1, engenheiros de software produzem código de produção através de rápidas interações como as que seguem:

1. O primeiro passo é adicionar um teste simples o qual é suficiente para a suíte de teste falhar.
2. Depois executamos nossa suíte para confirmar que os testes realmente estão falhando.
3. Agora atualize-se o código funcional a fim de passar no novo teste.
4. Executamos a suíte de teste para verificarmos se agora realmente passamos no novo teste.
5. Agora com o teste passando: são removidos as duplicações de código a fim de limpar o código.

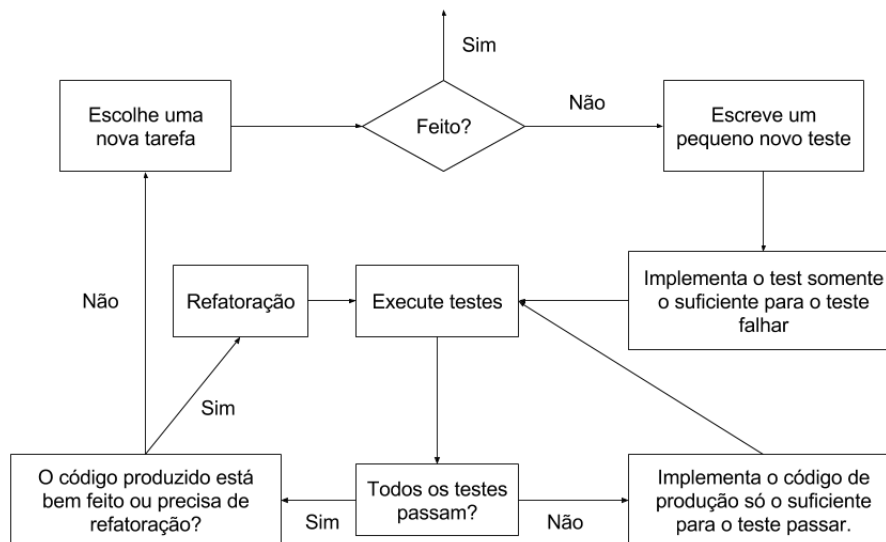


Figure 1: Fluxograma do TDD

3 Design de Software em TDD

O Test-driven Development permite que o desenvolvedor que está trabalhando um módulo, enquanto escreve os testes, pense como serão as responsabilidades, interfaces, quais serviços serão disponibilizados por esse módulo. Depois, enquanto escreve o código de produção, consegue definir um foco do que deve ser feito e mantém esse foco até que todos os testes passem. Fazendo assim, criamos um ritmo de código, teste até que todos os testes criados sejam implementados.

Em Test-driven Development, o código desenvolvido é mantido dentro do controle intelectual do desenvolvedor, já que o próprio escreveu os testes e ele ou ela está fazendo continuamente pequenas alterações de design e decisões de implementação, aumentando as funcionalidades do programa em um certo ritmo contínuo.

4 Objetivos ao se praticar TDD

TDD é uma prática de desenvolvimento que tem por objetivo garantir a qualidade e a confiabilidade do produto o mais rápido possível. Ao decorrer do desenvolvimento, todo o código elaborado é desenvolvido em conjunto com uma suíte de testes automatizados. Esses testes permitem uma segurança maior ao desenvolvedor quando precisa mudar algo que já foi implementado ou precisa refatorar o código.

Além disso, desenvolvedores experientes em TDD podem analisar se os testes produzidos estão difíceis de serem feitos e podem fazer refactoring ou mudanças.

5 Referências

Nachiappan Nagappan, E. Michael Maximilien, Thirumalesh Bhat, Laurie Williams, Realizing quality improvement through test driven development: results and experiences of four industrial teams. Disponível em <http://link.springer.com/article/10.1007/s10664-008-9062-z#/page-1>. Acessado em 26 de maio de 2016.

Andreas Augustin, Test-Driven Development: Concepts, Taxonomy, and Future Direction. Disponível em <https://www.semanticscholar.org/paper/Test-Driven-Development-Concepts-Taxonomy-and-Janzen-Saiedian/bdcd570eb6a45d7a9107a18e25f54b741b92177f/pdf>. Acessado em 26 de maio de 2016

Martin Fowler, Bill Venners: Test-Driven Development, A Conversation with Martin Fowler. Disponível em <http://www.biology.emory.edu/research/Prinz/Cengiz/cs540-485-FA12/resources/testDrivenDev.pdf>. Acessado em 30 de maio de 2016.