



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DO CEARÁ  
PRÓ-REITORIA DE ENSINO  
DEPARTAMENTO DE COMPUTAÇÃO DO CAMPUS MARACANAÚ  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**AUREA HELENA DE SOUZA GABRIEL PEIXOTO**

**PROCESSOS DE TESTES NAS METODOLOGIAS AGEIS - UMA  
REVISAO BIBLIOGRAFICA**

**MARACANAÚ  
2020**

ÁUREA HELENA DE SOUZA GABRIEL PEIXOTO

## PROCESSOS DE TESTES NAS METODOLOGIAS ÁGEIS

Monografia submetida à Coordenadoria do Curso de Bacharelado em Ciência da Computação do Instituto Federal do Ceará - Campus Maracanaú, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Área de pesquisa: Engenharia de Software

Orientador: Prof. Me. Fabiana Marinho

Maracanaú  
2020

Dedico este trabalho...

---

# Agradecimentos

---

Graças a Deus.

---

# Resumo

---

O presente texto representa o resultado de leituras realizadas a respeito da utilização das técnicas de teste utilizadas nos métodos ágeis, onde foi possível perceber que a presença de testes ágeis durante o processo de desenvolvimento de software contribui para o aumento da qualidade do produto final.

---

# Abstract

---

O presente texto representa o resultado de leituras realizadas a respeito da utilização das técnicas de teste utilizadas nos métodos ágeis, onde foi possível perceber que a presença de testes ágeis durante o processo de desenvolvimento de software contribui para o aumento da qualidade do produto final.

---

# Sumário

---

**Lista de Figuras**

**Lista de Tabelas**

**Lista de Abreviações**

<b>1</b>	<b>Introdução</b>	<b>10</b>
1.1	Contextualização . . . . .	10
<b>2</b>	<b>Uma pequena revisão bibliográfica</b>	<b>11</b>
<b>3</b>	<b>Considerações finais</b>	<b>14</b>
	<b>Referências Bibliográficas</b>	<b>15</b>

---

## **Lista de Figuras**

---



---

## **Lista de Tabelas**

---

---

# Lista de Abreviações

---

<b>TDD</b>	Test-Driven Development
<b>BDD</b>	Behaviour-Driven Development
<b>ATDD</b>	Acceptance Test-Driven Development

# INTRODUÇÃO

---

## 1.1 Contextualização

O processo de teste é uma fase de grande importância no ciclo de vida do software. O teste permite fazer uma análise do desenvolvimento do produto. É o momento de saber se o sistema está funcionando corretamente e se propõe a fazer aquilo que era esperado nos requisitos. A forma como os testes vêm sendo executados mudou com os passar dos anos. Nas metodologias tradicionais de desenvolvimento de software, ele é a última etapa do processo. Isso pode ser um problema pois se as especificações do produto não forem bem levantadas e compreendidas desde o princípio, pode ser preciso refatorar todo o código novamente. Atrasando a entrega do software, elevando o seu custo e aumentando a insatisfação do cliente.

Em metodologias ágeis, onde a maior prioridade é satisfazer o cliente com um produto final que atenda a suas necessidades, é incentivado o uso de técnicas de testes ágeis para um desenvolvimento com rapidez mas sem diminuir a qualidade. No eXtreme Programming (XP), por exemplo, foi introduzido o TDD, no qual um teste automatizado deve ser escrito antes da implementação do programa (test-first). Isto pode parecer um pouco complicado de fazer já que não há nenhum código funcional escrito ainda mas faz com que haja uma necessidade maior de se concentrar nos requisitos do produto. Os métodos de testes BDD e ATDD, por sua vez, foram desenvolvidos a partir dos princípios do TDD. Sendo criados para contornar algumas falhas de comunicação deixados por ele.

O presente texto busca fazer uma pequena revisão bibliográfica sobre as diferentes técnicas de testes usados nas metodologias ágeis. Aqui serão apresentados conceitos, características e aplicações dos testes TDD, BDD e ATDD.

## UMA PEQUENA REVISÃO BIBLIOGRÁFICA

---

Para [Palod & Chaus 2016] as metodologias ágeis buscam corrigir alguns problemas das ferramentas tradicionais de teste tais como dificuldade de manutenção do código. Isso acontece devido ao fato de que neste tipo de abordagem os testes são executados somente próximo ao final do projeto, após a escrita do código de produção.

O desenvolvimento guiado por testes (TDD) é uma técnica utilizada durante a fase de desenvolvimento de software onde os testes devem ser implementados antes do código funcional do sistema, ou seja, inclui os testes desde a fase inicial do projeto. A princípio pode parecer um pouco complicado mas ajuda os desenvolvedores a se concentrarem nos requisitos [Moe 2019]. A implementação do TDD obedece um ciclo chamado de vermelho-verde-refatora [Bulgareli & Foschini 2015]. Vermelho indica que o teste deve ser implementado e executado. Como não há nenhuma funcionalidade implementada ainda, o teste resulta em falha. Na etapa representada pelo verde deve ser escrito um código que passe no teste criado. Logo após, a refatoração serve para fazer uma análise da implementação e mudá-la caso necessário, objetivando sempre melhorar o código atual.

O desenvolvimento guiado por comportamento (BDD) surgiu da necessidade de explicar o funcionamento do teste de software em si. Para quem não sabe o que é um teste, como testar, quando testar e o que uma falha de teste significa, tentar explicar o TDD pode ser uma tarefa difícil [Bulgareli & Foschini 2015]. Pensando nisso, o BDD foi criado com intuito de fazer com que todas as partes interessadas no projeto pensassem no comportamento do sistema ao invés da sua implementação. Logo, a equipe de desenvolvimento deve criar testes baseados em cenários descritos por meio da análise da especificação antes de criar o código em si. Essa característica faz que todos os envolvidos consigam ter uma melhor compreensão dos requisitos. O BDD permite que pessoas da equipe que não tenham nenhum conhecimento técnico entenda melhor o processo de teste [Barus 2019]. Devido os testes serem escritos em uma linguagem simples, a curva de aprendizagem é mais curta (Moe, 2019). O

BDD segue o mesmo ciclo de funcionamento do TDD mas utiliza uma linguagem mais simples para criar as funções de testes.

O ATDD é muito similar ao TDD e ao BDD, pois é voltado para o lado do desenvolvedor mas toda equipe deve colaborar a escolher juntos os chamados critérios de aceitação antes do desenvolvimento do código. Tanto o BDD quanto o ATDD são abordagens que visam envolver o cliente em toda fase do ciclo de desenvolvimento de software [Barus 2019].

[Moe 2019] faz um estudo comparativo entre o TDD, BDD e ATDD. O autor descreve cada uma das metodologias de forma simples e resumida, lista as vantagens e desvantagens de se utilizar cada uma, além de pontuar as diferenças entre elas. Uma das características em comum nos três métodos é que há uma contínua execução de testes e refatoração de códigos. Por meio do artigo é possível também compreender que cada técnica é focada na perspectiva de um ou mais integrantes da equipe de desenvolvimento, sejam eles desenvolvedores, testadores ou clientes. E que elas foram criadas para suprir algum deficit de entendimento dos requisitos do sistema que possam afetar na qualidade do mesmo. Segundo Moe o TDD contribui para que os desenvolvedores foquem nos requisitos do produto enquanto o BDD e ATDD devido a sua linguagem não técnica permite uma melhor comunicação, colaboração e compreensão dos requisitos entre vários membros da equipe.

[Manuaba 2019] aborda a implementação do T-BDD, combinação do TDD com o BDD. Além disso o autor faz uma análise comparativa com o TDD. O sistema Vixio, uma plataforma web onde as pessoas podem escrever e compartilhar histórias de ficção interativas, foi utilizada com o intuito de fazer as comparações entre os dois tipos de testes citados. O estudo concluiu que ao combinar ambas as tecnicas obtêm-se uma melhoria nos testes. O uso das metodologias em conjunto obteve um ótimo desempenho atingindo uma alta cobertura de testes. Pode-se observar que houve problemas ao utilizar apenas a metodologia TDD a aplicação Vixio. Este método de teste não funcionou bem quando houve a necessidade de mudar os parâmetros de recursos do sistema, como por exemplo, modificar os dados do banco de dados. Enquanto o teste TDD apresentou falhas após a mudança no banco de dados, o T-BDD não retornou nenhum erro. [Barus 2019] no seu artigo discute a aplicação dos métodos ATDD e BDD para testar dois aplicativos web desenvolvidos por alunos do Instituto de Tecnologia Del, na Indonésia. Para aplicar o conceito de BDD e ATDD no projeto duas ferramentas de automação de teste foram utilizadas: Robot e Cucumber. O in-

tuito é descrever se os métodos de testes são eficazes e se ajudam a melhorar o desenvolvimento de software. Os resultados mostraram que os métodos são eficazes e auxiliam na eliminação de erros o mais rápido possível. A fase de experimentos consistiu na coleta de requisitos, na escrita do teste de aceitação, escrever o teste unitário com falha, executar o teste e refatorar o código. Durante as fases de testes foram encontrados alguns bugs em ambas as aplicações. Por meio do estudo o autor também pode concluir que ATDD e BDD envolvem o cliente em toda a fase do ciclo de vida de software e que os frameworks Cucumber e Robot funcionaram muito bem em conjunto com os métodos de testes que são foco do estudo.

## CONSIDERAÇÕES FINAIS

---

Conclui-se que os testes não servem somente para detectar e corrigir erros de implementação do código funcional. É uma fase também para saber se o sistema está funcionando de acordo com que foi levantado nos requisitos. E para que isso ocorra é preciso da participação de toda equipe, muito além de somente pessoas com conhecimentos técnicos. É preciso incluir pessoas de negócios nesta importante etapa. Os testes nas metodologias ágeis são usados com o intuito de manter toda a equipe de desenvolvimento do projeto focada e envolvida com os requisitos, através de práticas que possibilite uma maior compreensão das especificações do produto e aumente a comunicação entre todas as partes interessadas no projeto para que o produto final esteja de acordo com as necessidades do cliente, que é um dos princípios fundamentais da metodologia ágil (NGUYEN JENNY; VIDAL; AL-SHAMAA, 2020 apud BEEDLE *et al.*, 2001).

---

## Referências Bibliográficas

---

BARUS, Arlinta C. The implementation of ATDD and BDD from testing perspectives. 2019.

BEEDLE, M; BENNEKUM AV.AND COCKBURN, A; CUNNINGHAM, W; FOWLER, M; HIGHSMITH, J; THOMAS, D. Agile manifesto. 2001.

BULGARELI, Joao A; FOSCHINI, Ivan J. Requisitos testáveis com behaviour-driven development. 2015.

MANUABA, Ida BK. Combination of test-driven development and behaviour-driven development for improving backend testing performance. 2019.

MOE, Myint M. Comparative study of test-driven development (TDD), behaviour-driven development (BDD) and acceptance test-driven development (ATDD). 2019.

NGUYEN JENNY, Vidal; VIDAL, Daniel; AL-SHAMAA, Sawsan. Principles and practices of agile technique in automotive product. 2020.

PALOD, Heet; CHAUS, Danish. Agile testing automation framework. 2016.