



**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

**TÉCNICAS DE TESTES MAIS USADAS NAS METODOLOGIAS
ÁGEIS - UMA REVISÃO BIBLIOGRÁFICA**

**MARACANAÚ
2020**

ÁUREA HELENA DE SOUZA GABRIEL PEIXOTO

TÉCNICAS DE TESTES MAIS USADAS NAS METODOLOGIAS ÁGEIS - UMA REVISÃO BIBLIOGRÁFICA

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 texto aqui apresentado descreve o resultado inicial de uma revisão bibliográfica realizada sobre o uso das técnicas de teste nos métodos ágeis. De acordo com as leituras realizadas, 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. Na Seção 1, é descrito um contexto geral do trabalho. Na Seção 2, são discutidas as características das metodologias de testes usadas nos métodos ágeis e na Seção 3, as considerações finais e conclusões do estudo realizado são apresentadas.

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

1	Introdução	10
1.1	Contextualização	10
1.2	Objetivo Geral	11
1.3	Objetivo Específicos	11
2	Uma breve revisão bibliográfica	12
3	Considerações finais	15
	Referências Bibliográficas	16

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

TDD	Test-Driven Development
BDD	Behaviour-Driven Development
ATDD	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 e verificar se o sistema está correto com relação à sua especificação e atende às expectativas do cliente.

A forma de execução dos testes passou por mudanças na medida que novos modelos de processo e metodologias de desenvolvimento de software foram propostos. Nas metodologias tradicionais, o teste é a última fase do processo de desenvolvimento o que pode ser um problema, pois se as especificações do produto não forem identificadas e compreendidas corretamente desde o princípio, pode ser necessário revisar todas as fases, desde a eliciação dos requisitos até o código. Como consequência, gerar atraso da entrega do software, elevando o seu custo e aumentando a insatisfação do cliente.

Em metodologias ágeis, a maior prioridade é desenvolver um produto final que satisfaça às suas necessidades dos clientes. Nesse sentido, é incentivado o uso de técnicas de teste no início do processo de desenvolvimento, minimizando a possibilidade de retrabalho sem diminuir a qualidade. No método eXtreme Programming (XP), por exemplo, foi introduzida a metodologia Test Driven Development (TDD). Ao adotar a TDD, um teste automatizado deve ser escrito antes da implementação do programa, também denominado Test-First. Inicialmente, a TDD pode parecer complexa, já que ainda não existe código funcional escrito. No entanto, o uso da TDD exige uma maior concentração e melhor entendimento dos requisitos do produto.

As metodologias de testes Behavior Driven Development (BDD) e Acceptance Test-Driven Development (ATDD), por sua vez, foram desenvolvidas a partir dos princípios da TDD. Sendo criados para contornar algumas falhas de comunicação identificadas com a prática da TDD.

1.2 Objetivo Geral

O presente texto possui como objetivo geral, revisar por meio de uma pesquisa bibliográfica as técnicas de testes mais usadas nos métodos ágeis que foram abordadas na literatura.

1.3 Objetivo Específicos

Mostrar as diferenças entre as diversas técnicas de testes ágeis. Analisar como as técnicas de testes ágeis podem ser combinadas a fim de obter a melhoria da qualidade do teste de software. Identificar a aplicabilidade das técnicas de teste em projetos de software. Classificar em qual contexto de aplicação cada técnica melhor se adequa.

UMA BREVE REVISÃO BIBLIOGRÁFICA

Para [Palod & Chaus 2016] as metodologias ágeis buscam corrigir o principal problema existente nas técnicas tradicionais de teste que envolve a dificuldade de manutenção do código. Segundo os autores, isso acontece devido ao fato de que neste tipo de abordagem os testes são executados somente no final do processo de desenvolvimento, após a escrita do código de produção.

A TDD é uma metodologia de teste proposta nos métodos ágeis. De acordo com a TDD, os testes devem ser implementados antes do código funcional do sistema, ou seja, ela inclui os testes desde a fase inicial do projeto. A princípio, os conceitos dessa metodologia podem ser complicados, no entanto, ajuda os desenvolvedores a se concentrarem e compreenderem melhor os requisitos [Moe 2019]. A implementação da TDD obedece um ciclo chamado de Vermelho-Verde-Refatora [Bulgareli & Foschini 2015]. O 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. Em seguida, a etapa Refatora tem como objetivo fazer uma análise da implementação e mudá-la caso necessário, objetivando sempre melhorar a qualidade do código.

A BDD surgiu da necessidade de explicar melhor o funcionamento do teste de software. Explicar a metodologia TDD para programadores que não conhecem o conceito de teste, como testar, quando testar e o que uma falha de teste significa, pode ser uma tarefa difícil [Bulgareli & Foschini 2015]. Pensando nisso, a BDD foi criada com intuito de fazer com que todas as partes interessadas (não apenas os programadores) pensassem no comportamento do sistema ao invés da sua implementação. Dessa forma, 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.

A BDD permite que pessoas da equipe que não tenham nenhum conhecimento técnico entendam 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]. A BDD segue o mesmo ciclo de funcionamento da TDD, mas utiliza uma linguagem mais simples para criar as funções de testes.

A metodologia ATDD é similar à TDD e à BDD, pois não é voltada apenas para os desenvolvedores, mas toda equipe deve colaborar e definir os chamados critérios de aceitação antes do desenvolvimento do código. Tanto a BDD quanto a 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 a TDD, a BDD e a 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. Segundo Moe, uma característica comum nas três metodologias é que os testes são executados e o código é refatorado de forma contínua. Além disso, Moe enfatiza 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 o déficit de entendimento dos requisitos que pode afetar a qualidade do produto final. Ainda de acordo com Moe, a TDD contribui para que os desenvolvedores foquem nos requisitos do produto, enquanto a BDD e ATDD, devido à sua linguagem não técnica permite uma melhor comunicação, colaboração e compreensão dos requisitos entre vários membros da equipe.

[Manuaba 2019], aborda a implementação da T-BDD, uma combinação da TDD com a BDD. Além disso, o autor descreve uma análise comparativa da T-BDD com a TDD. O sistema Vixio, uma plataforma web na qual as pessoas podem escrever e compartilhar histórias de ficção interativas, foi utilizada para que a realização da análise comparativa entre a T-BDD e a TDD. O estudo conclui que, ao combinar ambas as técnicas, obtêm-se uma melhoria nos testes. O uso das metodologias em conjunto obteve um ótimo desempenho atingindo uma alta cobertura de testes. Outro resultado obtido com esse estudo é que foram observados problemas ao utilizar apenas a metodologia TDD no teste da aplicação Vixio. A TDD 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 os conceitos da BDD e da ATDD no projeto duas ferramentas de automação de teste foram utilizadas: Robot e Cucumber. O intuito desse estudo está em verificar se as metodologias de teste BDD e ATDD são eficazes e se ajudam a melhorar o desenvolvimento de software.

Os resultados obtidos foram positivos, pois mostraram que as metodologias 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, na escrita do teste unitário com falha, na execução do teste e na refatoração do código. Durante as fases de testes foram encontrados alguns bugs em ambas as aplicações. Por meio do estudo, o autor também concluiu que a ATDD e a BDD envolvem o cliente em toda a fase do ciclo de vida de software e que os frameworks Cucumber e Robot funcionam bem em conjunto com as metodologias de testes que são foco do estudo.

CONSIDERAÇÕES FINAIS

A revisão bibliográfica realizada neste estudo obteve como conclusão que os testes não devem ser utilizados apenas para identificar e corrigir erros de implementação do código funcional. A fase dos testes também pode ser executada para verificar se o sistema está funcionando de acordo com os requisitos do cliente. Para que isso ocorra é necessária a participação não só da equipe técnica, como também 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.